

2012 Buyers' Guide Inside!

MacTech Magazine
December 2011 • Issue 27.12

MACTECH[®]

The Journal of Apple Technology

LEAVES OF DATA

**Branching into Data Trees
in AppleScriptObjC**

**An Introduction
to GNU Octave**

**Object Orientation:
Properties and
Other References**

MACTECH.COM

\$8.95 US, \$12.95 Canada



ISSN 1067-8360 Printed in U.S.A.

Pioneer



AppRadio™

A revolutionary way to use apps in the car

RETHINKING CONNECTIVITY

www.pioneerelectronics.com/appradio

PIONEER is a registered trademark of Pioneer Corporation. AppRadio is a trademark of Pioneer Corporation. ©2011 Pioneer Electronics (USA) Inc.
Limited apps are compatible with this model. Certain functions such as video playback and keyboard input are only available while parked.
For more information, see www.pioneerelectronics.com/appradio.

LIVE CODE
Game Academy
December 6th to January 31st



Learn how to make killer games

Video tutorials
Get sample game code
Access expert forums and support staff

WINNER
OF THE MACTECH
SELECT AWARD FOR
BEST DEVELOPER TOOL

***QUOTE COUPON MACTECHGAME AND GET 20% OFF
LIVECODE IN THE RUNREV STORE.**

runrev



see our game in action
Download the Galactic Gauntlet through iTunes

LIVE CODE

SIGN UP TO OUR GAME ACADEMY BY VISITING WWW.RUNREV.COM/ACADEMY

*20% OFF LIVECODE GAME ACADEMY LICENSE FOR ANDROID OR IOS, PERSONAL OR COMMERCIAL EDITION

**HEADPHONES FOR
CRITICAL LISTENING.
AND
CRITICAL LISTENERS.**



SRH240



SRH440



SRH840



SRH940
NEW

Recording. Mixing. Monitoring. Whatever you're up to, Shure Professional Headphones are up to the challenge. With even, accurate sound across an extended frequency range, they deliver everything you need to get it right. And a lightweight design will keep you listening comfortably from the first take to the last. Learn more at Shure.com.

www.shure.com

©2011 Shure Incorporated

SHURE®
LEGENDARY
PERFORMANCE™

TABLE OF CONTENTS

Mac in the Shell

Brewing Up Missing Apps

Using homebrew to install additional apps on OS X.

by Edward Marczak. 6

Object Orientation

Properties and Other References

Objects knowing about other objects

by Peter Hosey. 16

An Introduction to GNU Octave

Is Octave a viable MATLAB alternative?

by Mihalis Tsoukalos. 32

Leaves of Data

Presenting data trees with AppleScriptObjC

by José R.C. Cruz 44

MacEnterprise

Apple Software Updates with Reposado

An open-source alternative to Apple's Software Update service

by Greg Neagle, MacEnterprise.org 56

2012 Buyers' Guide

Product Guide: Listing by Company 62

Product Guide: Listing by Category 67

Product Guide: Listing by Product 68

The MacTech Spotlight

Guy English

<http://kickingbear.com>. 72

From the Editor

The last month of the year always inspires the desire to look back and see where we've been. It's probably useful to do that individually, certainly, but also as a community. 2011 brought both incredible highs and lows. We saw the release of the iPad 2, iOS 5, OS X Lion along with new hardware that seems to keep exceeding people's expectations. Of course, this all came along with the loss of Steve Jobs, the co-founder and public personality of Apple. We saw Apple's stock peak at \$422.24, which, at some point in the past, seemed impossibly out of reach. But then, Apple routinely makes the seemingly impossible, possible.

Of course, coming up on a new year inspires us to look forward. Now, I'm not one generally for New Year's resolutions, but it happens to be a great practice for your tech-self. Make a big resolution, or a small one. Revamp your entire application. Learn a new language. Better learn your text editor. Learn a new command for your source control. Learn a new shell command. Any one new thing, just pick one and follow through.

We're going to help you on this path, of course. Get an early jump on your resolution, starting with this month.

This month's Mac in the Shell guides you through one way of getting even more shell commands onto your Mac so you have more to choose from to learn. The introduction to Homebrew is just the beginning of what you can get out of a source-based package manager.

Regular (and frequent) contributor Mihalís Tsoukalos show you how to use Octave, a specialized open source math package that can replace MatLab. If you have the need for something more powerful than bc or Excel, check out Octave.

If you're looking to learn a new language, you should be following Peter Hosey's series on Objective-C. Peter has covered the basics of C, and built up to the extensions that make up modern Objective-C. This month, he expands on the object hierarchy and how objects know about one another.

To expand your horizons beyond a pure OS X infrastructure, Greg Neagle's Mac Enterprise column covers Repasado, a way to run a Software Update Server without OS X...on any platform that supports Python.

Naturally, we can all learn from our peers, so check out what Guy English has to say, as he's featured in this month's MacTech Spotlight. Guy wears many 'hats' in the Mac community and brings great perspective to his work.

Finally, no matter your tech-resolution, we want to help out. Let us know what you're looking for at letters@mactech.com. See you in 2012!

Edward Marczak,
Executive Editor



Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service

cust_service@mactech.com

Press Releases

press_releases@mactech.com

Ad Sales

adsales@mactech.com

Editorial

editorial@mactech.com

(Authors only, no pr)

Accounting

accounting@mactech.com

Marketing

marketing@mactech.com

General

info@mactech.com

Web Site

http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail**?

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

MACTECH[®]

The Journal of Macintosh Technology

A publication of **XPLAIN** CORPORATION

The Magazine Staff

Publisher & Editor-in-Chief: Neil Ticktin

Executive Editor: Edward R. Marczak

Business Editor: Andrea Sniderman

Ad Director: Bart Allan

Production: David Allen

News: Dennis Sellers

Podcast Producer: Josh Long

drupalmaster: Erik Peterson

Xplain Corporation Senior Staff

Chief Executive Officer: Neil Ticktin

President: Andrea J. Sniderman

Accounting: Marcie Moriarty

Customer Relations: Susan Pomrantz

Columnists

Mac In The Shell: by Ed Marczak

Swaine Manor: by Michael Swaine

KoolTools/Geek Guides: by Dennis Sellers

MacEnterprise: by Greg Neagle and Philip Rinehart

Greg's Bite: by Greg Mills

Regular Contributors

José R.C. Cruz, Michael Göbel, Michele Hjörleifsson, Mihalis Tsoukalos

Oliver Pospisil, Rich Morin, William Smith

Canada Post: Publications Mail Agreement #41513541

Canada Returns to be sent to: Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 705 Lakefield Road, Suite 1, Westlake Village, CA 91361. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are \$47.00 per year. Canadian subscriptions are \$59.00 per year. All other international subscriptions are \$97.00 per year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

Opinions expressed are not necessarily the views of MacTech Magazine or Xplain Corporation. All contents are Copyright 1984-2011 by Xplain Corporation. All rights reserved. MacTech is a registered trademarks of Xplain Corporation. MacNews, Xplain, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, Apple Expo, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

MAC IN THE SHELL

by Edward Marczak

Brewing Up Missing Apps

Using homebrew to install additional apps on OS X.

Introduction

OS X is really an incredible system: a beautiful GUI, and a powerful Unix subsystem. Apple includes a wealth of tools right out of the box. From scripting languages like Ruby and Perl to familiar Unix utilities like `tidy` and `awk` to Apple-specific command-line tools like `sips` and `textutil`. Even with all of this power, there are other tools available. As the technology craftsman, you need to choose the best tool for the job. That's where third-party package managers come in: to easily install all of the software that you need that Apple didn't include.

Utopia (it isn't)

What else could you possibly want, right? OS X includes everything you'd ever need! As it turns out, there are a lot of useful utilities. Like all good Unix systems, this software is just waiting for you to install. On most new machines, there are certain things I'd like to have available and set up, and a package manager is just the tool that will help get this done.

As a quick aside, package management is difficult. Apple keeps making steps in the direction of putting a real, full package manager on OS X, but hasn't ever really went all the way. Hence the need for third-party tools. There have been several attempts over time. Fink (<http://www.finkproject.org/>), is really the oldest of these, and is up to date for Lion. MacPorts (néé DarwinPorts, available from <http://www.macports.org/>) has also been around for a bit. The newest package manager, and subject of this article, is Homebrew (<http://mxcl.github.com/homebrew/>). Also worth mentioning is Munki (<http://code.google.com/p/munki/>), which was

written about in the October 2010 issue of MacTech Magazine. Munki is a package manager where you control the catalog available for your fleet. Compare that to Fink, Macports and Homebrew, where you pretty much rely on package maintainers to create Mac-specific ports of code that the package manager can install. Perhaps Apple's real package manager is the AppStore? (OK, it's largely an incomplete package manager, but there's potential there.)

All three major package managers—Fink, MacPorts and Homebrew—stay out of the way of stepping on system binaries and make for easy removal.

Installing Homebrew

I've written about Macports in the past, but wish to explore an alternative this month: Homebrew. Homebrew is a relatively new third-party package manager for OS X, and I'll be focusing on the experience on Lion. This is a great time to jump into Lion, as 10.7.2 has cleared up a number of issues, along with the latest version of Xcode (4.2.1 as of this writing), which also includes `llvm` and `clang` options. You'll need Apple's Developer Tools installed prior to running any Homebrew commands. So, if you need to, go to the AppStore and install the Xcode installer. (That's right: the Mac AppStore doesn't install Xcode, but rather, it installs an Xcode installer, which you then need to run to actually install Xcode. Ensure this is all complete before proceeding.) Next, run the Homebrew installer:

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.github.com/gist/323731)"
```

(See the Homebrew home page linked above for more information about this installer and script.)

Another aside: Homebrew, as you'll see, is a little different from other package managers. By default, Homebrew sets itself up in a way that requires the id installing it be in the admin group. Additionally, it will chown `/usr/local` to the id of the person installing it. If these are deal-breakers—and they very well may be—other options include installing in your home directory by expanding an archive, and/or not installing/using it at all. (Seriously, if you can't reconcile the security implications with yourself, don't use it. I'll claim this for any software.)

When you run this script, and it runs its initial checks, it will alert you and request permission for the changes it is about to make. Once it completes, run `brew update`. Brew's update command populates the git repository that Homebrew uses for all of its brews. (Xcode now includes git, which is one reason why you need to install Xcode prior to running the brew command. Interestingly, installing git is

Instant iOS Device Management



MaaS360[®]
by Fiberlink

Cloud-based MDM in under 5 minutes. No servers needed.

Try it now @ www.maas360.com/MacTech



one of the main reasons I used to get a package manager like Homebrew installed.)

Ready For Use

Once the Homebrew package manager is installed, you're ready to use it to install the software you really want. You'll need Internet connectivity anytime you need to fetch a package, and possibly its dependencies, from scratch. Let's start with something simple: ctags. Specifically, exuberant ctags ("A multilanguage implementation of ctags"). Apple does include ctags in OS X, but it's a bit anemic. Let's see if Homebrew can help out: does it have a port for this package?

```
$ brew search ctags
ctags
```

Yes! Let's get some more info and ensure it's the right one:

```
$ brew info ctags
ctags 5.8
http://ctags.sourceforge.net/
Not installed
http://github.com/mxcl/homebrew/commits/master/Library/Formula/ctags.rb
```

Excellent. Now, to install it, it's a simple matter of asking Homebrew to install it:

```
brew install ctags
```

Homebrew will download, compile and install ctags under /usr/local/bin.

Now, one of the only reasons I ever install Exuberant ctags is for integration with Vim. Not just any Vim, though: it's MacVim I'm after. I used to just grab the binary distribution from GitHub, but the way it's currently configured makes that a bit of a pain to automate. However, like Vim, MacVim is open source and available as a package in Homebrew:

```
$ brew search macvim
macvim
```

Now, if you know a bit about MacVim, you can get different behavior depending on how you compile it. You may want a very small, stripped down version of it, or, you may want to include everything it offers. Homebrew can tell you the pre-defined options that a package has using the **options** command:

```
$ brew options macvim
macvim
-custom-icons
  Try to generate custom document icons.
-with-cscope
  Build with Cscope support.
-with-envycoder
  Build with Envy Code R Bold font.
-override-system-vim
  Override system vim.
-enable-clipboard
  Enable System clipboard handling in the terminal.
```

Got Mac® mini servers?



Rack 'em Right

RackMac™ mini secures up to two Mac mini servers side-by-side in a rugged 1U rackmount enclosure. Much more than just a rackable shelf, key features include airflow management for optimized cooling, front panel access to power switches, USB ports and SuperDrives for convenient control, plus easy cable management in the back. RackMac mini rocks!

SONNET
SIMPLY FAST™



Easy Access. Accessible front panel power buttons & USB ports.

Stay Cool. RackMac mini's specially designed chamber directs cool air through front panel openings to the computers' air intakes, and then out the back.

sonnettech.com/rackmacmini

©2011 Sonnet Technologies, Inc. Sonnet, the Sonnet logo, Simply Fast, and RackMac are trademarks of Sonnet Technologies, Inc. Mac and SuperDrive are a trademark of Apple, Inc., registered in the U.S. and other countries. Thunderbolt and the Thunderbolt logo are trademarks of Intel Corporation in the U.S. and/or other countries.

25
Years of Simply Fast
Innovations

THUNDERBOLT.
Learn more about
our upcoming Thunderbolt
technology products
[sonnettech.com/
thunderbolt](http://sonnettech.com/thunderbolt)

WORLDS FIRST

USB TOKEN TO MAKE EMAIL SERVICE HACKING PROOF!

SAFE - SECURE HACKING PROOF



Features:

Two factor Security to make email account hacking proof
Portable and secure email from any browser
True end-to-end security - using 32+ Antivirus engine scanning
Anonymous email headers - Hide ur IP for security
Windows, Linux and Mac OS support
Use your own domain name and previous email id

The Secure Mail Key technology protects effectively and conveniently against:

- Any type of Rootkits/Trojans/Keylogger
- Man-in-the-browser attacks
- Man-in-the-middle attacks
- Session hijacking
- Real time phishing attacks

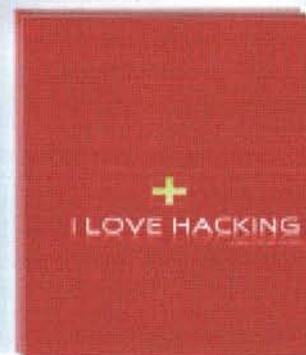
We can Deploy this service on:

Our own Secure server (webmail)
Google app
Your email server

For more information and a Free Trail or to Order
visit us at : www.securemailkey.com

Other Services:

- * Advanced Ethical Hacking books
Metasploits, Rootkit/Botnet development,
Exploit development, Website/server hacking
- * Training on Ethical Hacking
- * Antivirus with Anti Execution Technology
- * IT security consulting, Hire a Hacker for Pen testing



Swiss Hacker

Swiss Hacker GmbH | P.O. Box 392 | 8058 Zurich - Airport | Switzerland +41 435081415

616 Corporate Way, Suite 2 #4000 | Valley Cottage | New York 10989, USA | 818 252 9090

Website: www.swisshacker.com

email: office@swisshacker.com

There's one more option built into Homebrew that I'm going to use here: `—HEAD`. The `—HEAD` option builds from the latest check-in. So, our command looks like this:

```
brew install macvim --enable-clipboard --with-envycoder --HEAD
```

At the end of the MacVim install, you'll be told that since this run produces a binary that conflicts with a system binary (vim), that it's not put into `/usr/local/bin` immediately. You're instructed to type 'brew link macvim' to link the binary into `/usr/local/bin`, and you should likely do so now. I also go one step further and link the MacVim directory into my home Application directory:

```
ln -s /usr/local/Cellar/macvim/HEAD  
$(HOME)/Applications/MacVim
```

(You may need to create `~/Applications` first.) This allows me to easily access the MacVim GUI app.

There are hundreds of packages available, but there's one in particular that I like to have on a system that I use: Wireshark. If you've been wanting to follow along with some of the Wireshark articles that MacTech Magazine has been running, here's your chance to easily get set up. Let's look at doing this now.

First, a `brew search` tells us that Wireshark is available. A `brew options` command tells us that there is one option: `—with-x`. Since this is what compiles up the GUI, you want this option. I'm going to add in some further options.

First, I like more detail in what's actually happening when, so I include `-vd`, for "verbose, debug." Next, I want to take advantage of the new, powerful capabilities of `llvm` and `clang`. So I add `—use-llvm` and `—use-clang`. The final command looks like this:

```
brew install -vd wireshark --with-x --use-llvm --use-clang
```

You can now watch (or, get a beverage) as Homebrew downloads and compiles Wireshark along with its many dependencies. Really, it takes a while. However, you end up with a working, optimized version of Wireshark on your machine.

How Did I Get Here?

Homebrew relies heavily on git and having a clean git repository of the packages it offers. Each formula that Homebrew uses to know what to do is a Ruby script. In this manner, you can alter what's happening, fix issues with

Smart Strip

The Control Outlet affects all the Automatically Switched Outlets.

- Saves an average of \$52.91 a year in energy costs.
- Recommended as one of the top 5 ways to save energy!
- Pays for itself in as little as 6 months.



Going GREEN

BITS LIMITED

made easy!

BITS LIMITED

For more information you can contact us at:

813-425-5744
sales@bitsltd.net
www.bitsltd.net

True Surge

Truly protects your equipment from power surges!

- The only daisy chainable surge protector on the market.
- Totally disconnects your equipment in the event of a power surge.
- Safest surge protector available!



Charging Station

When the batteries in your mobile devices are fully charged, it automatically unplugs them.

- Saves up to \$34.33 a year in energy costs.
- Stop killing your batteries by overcharging them!
- More than doubles the lifespan of your mobile devices.



powersquid®

No more wasted outlets. Great for the workshop.

- Very durable and versatile.
- No more damaged plugs on your equipment.
- Flexible arms make it easy to plug & unplug standard or transformer plugs.
- Recessed lighted power switch eliminates accidental shutoff.



6 Bank Radio Packages Starting At

\$1,399

Radio Rentals Starting At

\$1.99 Per Day



Dollar RADIO RENTALS & SALES

www.dollarradiosales.com

888-968-9693

www.dollarradiorentals.com

866-886-3767

outdated or otherwise non-working formulae, or, contribute your own.

If you're interested in creating your own Homebrew formula—either for inclusion in the Homebrew collection at large, or for simple internal distribution—you use the create command:

```
$ brew create http://example.com/foo-0.1.tar.gz  
Created /usr/local/Library/Formula/foo.rb
```

The URL here is the URL to the downloadable source.

The great thing about a shell tool is that it can be automated. I switch and move between machines pretty often. I have a setup script that configures a given Mac for me in just the way I want it. Part of my script simply does all the things mentioned in this article: it checks for Xcode and if present, runs the Homebrew installer, installs the packages mentioned and links them in place. (In reality, there are a few other packages I install, including gfortran and R.)

Troubleshooting

A deep guide to troubleshooting Homebrew is a little outside the scope of this article. However, just keep this in mind: don't panic. If anything in Homebrew fails, you do *NOT* damage your system. All of the OS X tools are still intact. In a worst case scenario, you could remove

Homebrew and reinstall, although that's typically not necessary.

Before installing any new package, you should run **brew update** to ensure that you have the latest formulae. Even having done that, things on this big, crazy Internet change daily, and a site that once offered a download may no longer offer it or just temporarily be down. You can inspect what a formula is doing by issuing 'brew edit [formula]'. To see how simple a given formula can be, try **brew edit browser**. For something more practical, try **brew edit etl**. In the latter example, it's pretty clear where the formula expects to download the source from (the 'url' parameter) and how to go about installing it (the 'install' method). Does the source download for you manually? (Use **curl** or your web browser.) Does a **brew update** resolve the issue?

Failing to resolve your issue with those tips, there's one that's almost sure to work: a Google search. Homebrew has an active community and you're likely not the first one to notice the problem. Very often, this leads you to the Homebrew tracker where someone has already posted a solution.

Conclusion

Apple makes pragmatic choices as to what to include in the base OS. It strikes a great balance between common utilities and bloating the system with the kitchen sink. Now

CONVERT DVI TO MINIDISPLAYPORT

Gefen introduces a new solution for enabling computers with DVI connectors to utilize new Apple displays using the MiniDisplayPort connection. The converter is a low cost solution available that makes the legacy computers useful for the foreseeable future.



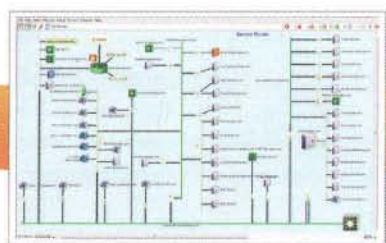
www.gefen.com

Stretch It. Switch It. Split It. Gefen's Got It.®



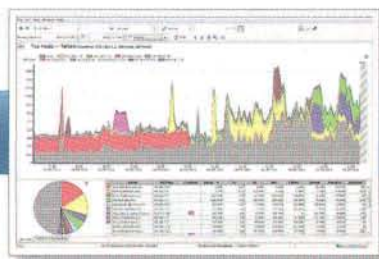
The #1 Network Monitoring, Mapping and Alerting Software for the Mac

Diagnostics



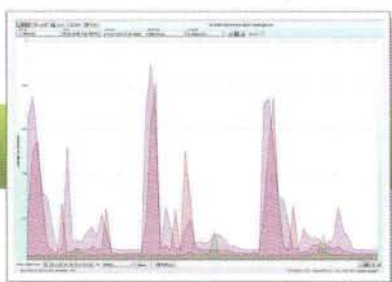
Layer 2 Connectivity

Troubleshooting



Flows Analysis

Capacity Planning



Reporting

- **Powerful**—Monitor anything connected to your network
- **Flexible**—Runs on Windows, Mac, Linux
- **Easy to use**—Simple configuration with auto-discovery
- **Affordable**—Fully integrated, no add-ons to buy



Download a Free 30-day Trial
www.intermapper.com

WATCH THE WATCHER!

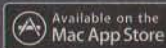
SPY CAM IS YOUR OWN PERSONAL SPY!

The perfect application for private video-surveillance.
With Spy Cam, turn your Mac into a surveillance camera.

Spy Cam by **AliceDev.com**



- Can be set up in 1 minute, all ready to go.
- Choose the recording interval you prefer: every 1/3/5/10/15/20/25/30 minutes.
- Set the video clip length: 5/10/15/20/30/60 seconds.
- Obeys a single master: you only need to password-protect it.
- Automatically sends videos to your Dropbox account.
- Costs 10 times less than a conventional video surveillance system!



IT KEEPS AN EYE ON YOUR MAC 24/7: SILENTLY AND COVERTLY!

that you know how to add the tools you need, there's no reason you can't be the most effective at your craft. Homebrew makes it easy to install those extra tools, or custom versions of them, to boost your productivity. If you don't choose Homebrew, I encourage you to check out Fink and MacPorts as ways to add those missing applications to your system.

Media of the month: "The Pragmatic Programmer: From Journeyman to Master," by Andrew Hunt and David Thomas. Even if you're not a "programmer," give this a read. Seriously. The tips, tricks and discipline it lays down is applicable to anyone in computer tech.

Until next month, keep brewing!

MI



About The Author

Edward Marczak is the Executive Editor of MacTech Magazine and helped co-found the annual MacTech Conference. He holds a place on the Mac team at Google and is a husband, father and general nerd.

BMS

**THE LAW OFFICE OF
BRADLEY M. SNIDERMAN**

Helping clients with their software legal issues.

- **Trademark and Copyright Registration**
- **Trade Secret Protection**
- **Licensing and Non Disclosure Agreements**
- **Assist with Software Audits**

I am an attorney practicing in Intellectual Property, Business Entity Formations, Corporate, Commercial and E-commerce Law.

Please give me a call or an e-mail. Reasonable fees.

23679 Calabasas Rd. #558 • Calabasas, CA 91302
PHONE 818-706-0631 FAX 818-332-1285 EMAIL brad@sniderman.com

iPAD PUBLISHING

FOR EVERYONE



SPECIAL OFFER: **ALL UPGRADES ONE PRICE**

Upgrade from **any version** to QuarkXPress 9 for only **\$299**

With **QuarkXPress 9** and App Studio, everyone can create apps and publish to the iPad!

Easy, affordable, innovative - iPad publishing for everyone.

Discover App Studio today at www.quark.com/AppStudio.



OBJECT ORIENTATION

by Peter Hosey

Properties and Other References

Objects knowing about other objects

Previously...

Last month, I showed how you lay out objects in a directed graph that conforms to the Model-View-Controller pattern, which clarifies both the design of your application and the ownership graph of its objects.

That covers the nodes, but what about the edges? How do you actually connect these objects together?

That's what I'll be showing this month, as I go in depth into instance variables, properties, and outlets.

Instance variables

As you saw in one of the earlier installments in this series, a class can have any number of instance variables. You declare them in the instance-variables section at the top of that class's `@implementation`:

```
@implementation Person
{
    //Objective-C instance-variable declarations use the same syntax
    as C variable declarations.
    NSString *name, *postalAddress, *emailAddress,
    *telephoneNumber;
    NSImage *photograph;
}

@end
```

Within any instance method, it's possible to access an instance variable of any instance of the same class; the syntax is `object->variable` (where `->` is C's pointer-to-member operator, otherwise only used with pointers to structures). Thus, to access an instance variable of `self`, you can say `self->variable`. As a convenience, the compiler lets you use the variable name alone, leaving the "`self->`" implied.

Accordingly, the only place where you can access instance variables by name alone, without specifying the instance, is within instance methods in the class's `@implementation`. This is because only an instance method has the implicit `self` argument referring to an instance.

Class methods have `self`, but referring to the class; C functions don't have `self` at all (at least not as an implicit argument). With no dominant instance, it's not clear what object you want to access an instance variable of.

You declare instance variables within a class implementation, but they are *instance* variables; the variables belong to each instance, not to the class. The class has no variables of its own. The declarations simply state what variables each instance will have.

Every instance gets its own set of instance variables that follow the declarations in its class and superclasses. Each instance's variables are part of it, so setting an instance variable in one instance does not make the value show up in the variable of the same name in any other instances.

Since instance variables are part of the instance, they last as long as the instance, and each variable holds its value until the instance is deallocated or the variable gets a new value assigned to it.

When an instance is created, all of its instance variables are initialized to `nil`. (Contrast with local and global variables, which aren't initialized by default.) This makes it easy to "lazily" create any other objects that the new object will need: Whenever it needs an object that you keep in one of its instance variables, you can test whether you have created that object yet by testing whether the variable's value is `nil`; if it is `nil`, then you have not created the object yet, so you create it and store its pointer in the variable for next time.

(Quick reminder before I continue: I've started assuming that you're using ARC. If you insist on sticking with manual reference-counting, see the Advanced Memory Management Programming Guide in Apple's documentation.)

Instance variables that hold pointers to other instances are ownerships of those objects. As long as you hold an object's pointer in your instance variable, you'll be keeping that object alive. Accordingly, you can release that ownership by assigning a different pointer (or `nil`) to the variable.

Any ownerships that an object holds on other objects are automatically released when the owning object is deallocated.

A variable can be declared with an "ownership qualifier" on its type. For most variables, this means very little. For instance variables, however, the ownership qualifier is important whenever you need behavior different from the default.

The default behavior is for every instance variable that points to an object to implicitly be `__strong`, which means that it is an ownership.

On Mac OS X 10.7 and later and iOS 5.0 and later, instance variables that point to objects can be declared with the `__weak` keyword. That keyword has two effects: first, the variable will not establish an ownership of the object; second, if the object is deallocated, any `__weak` variables that point to it will automatically be set to `nil`, saving you from subsequently trying to talk to a dead object (which would usually cause a crash).

Blackmagicdesign



The World's first Thunderbolt™ based capture and playback for SD/HD-SDI, HDMI and analog video!

Built on revolutionary Thunderbolt™ technology, UltraStudio 3D has a blazingly fast 10 Gb/s connection that's up to 20 times faster than USB 2.0! Machined from a solid piece of aluminum, UltraStudio 3D is an attractive, rugged device that's packed with features previously only available with a workstation. It's perfect for those on the go as an extremely portable companion to your camera, notebook and favorite editing software.



Connect to any Deck, Camera or Monitor

UltraStudio 3D features a huge range of video and audio connections. Dual Link 3 Gb/s SDI, HDMI, component analog, NTSC, PAL and s-video for capture and playback in SD, HD or 2K. Also included is 2 ch XLR AES/EBU audio and 2 ch balanced XLR analog audio. Connect to HDCAM SR, HDCAM, Digital Betacam, Betacam SP, HDV cameras, big-screen TVs and more. UltraStudio 3D even supports two streams of full resolution video up to 1080p HD for new stereoscopic 3D workflows!



Hardware Down Conversion

If you've ever wanted to monitor in both HD and SD while you work, then you'll love the built in high quality down converter. Use the Dual Link SDI outputs as a simultaneous HD and SD output, or you can switch back to Dual Link 4:4:4 when working in the highest quality RGB workflows. Select between letterbox, anamorphic 16:9 and even center cut 4:3 down conversion styles!



Advanced 3 Gb/s SDI Technology

With exciting new 3 Gb/s SDI connections, UltraStudio 3D allows twice the SDI data rate of normal HD-SDI, while also connecting to all your HD-SDI and SD-SDI equipment. Use 3 Gb/s SDI for 2K and edit your latest feature film using real time 2048 x 1556 2K resolution capture and playback!



More Third Party Software Compatibility

UltraStudio 3D works with the software you love to use! Use QuickTime™ software, or the world's most popular editing software such as Final Cut Pro™ and Premiere Pro™! You also get Photoshop™ plug-ins to grab and output frames, plus real time preview in After Effects™ and Nuke™. No other editing solution supports more software on Mac OS X, so now you have the freedom to build your studio your own way!



UltraStudio 3D
\$995

Learn more today at www.blackmagic-design.com/UltraStudio3D

TAKE YOUR CRASHPLAN TO THE NEXT LEVEL

Let our experts show you how with
our enhanced services and solutions.

- All-in hosting of offsite/slave CrashPlan PROe servers
- Extended monitoring, alerting and trend reporting options
- Provisioning and billing solutions for schools, enterprises and managed backup service providers

Request a free trial at
BackgroundBackup.ca



BACKGROUNDBACKUP

The CrashPlan PROe logo is a registered trademark of Code 42 Software, Inc., in the U.S. and other countries. All rights reserved. © Copyright 2011, Background Backup, Inc.



CRASHPLAN PROe
MASTER INTELLIGENCE



The third ownership qualifier for instance variables is `__unsafe_unretained`. This is like `__weak`, in that it does not establish an ownership (hence “unretained”), but without the automatic-set-to-nil part: if the object dies, this property will continue to point to it, leaving it possible to send a message to a dead object (hence “unsafe”). You generally should not use this keyword.

There is a fourth ownership qualifier, `__autoreleasing`, but that's only for local variables, so I won't talk about it here. Like all of these, it's covered in detail in the “Transitioning to ARC Release Notes” in Apple's documentation, and in even more detail in the ARC specification on the Clang project website.

Properties

A property formally expresses a relationship to a value or another object.

You declare a property within an `@interface` or `@protocol`. A property declaration looks like this:
`@property(keywords) type name;`

The latter half of this will look very familiar: it's the same as a variable declaration. As with a variable, this declares what type of value the property holds and the name of the property.

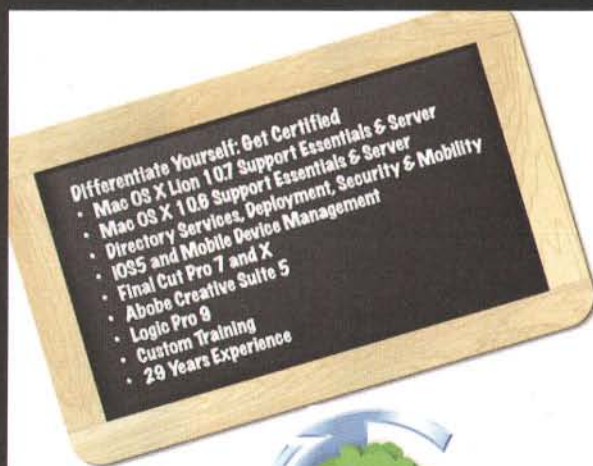
A property doesn't, strictly speaking, declare a variable, though. What it does declare is one or two methods, which are the accessors of the property. You send the object one of those messages to access the property.

Declaring a property obliges you to implement it in your class's `@implementation`. You have several options:

- `@dynamic propertyName;` is the hand-wave implementation. When you use `@dynamic`, you tell the compiler that the implementation will be provided by a superclass, at run time, or both. You mainly use this when inheriting the implementation from a superclass, such as when you subclass Core Data's `NSManagedObject`. Most of the time, this is not what you want.
- `@synthesize propertyName;` tells the compiler to generate an instance variable and an implementation of the property's accessor method(s). When you don't need your accessors to do anything unusual, this is the best choice.
- You can implement either or both accessor methods yourself. When you do this without synthesizing, you need to declare an instance variable yourself or create another place in which to store the value of the property.
- You can both synthesize the property *and* implement either or both methods yourself. The main reason to do this is so you can use the synthesized instance variable from your custom accessor method(s), but do something unusual in addition to the access. Most commonly, you'll do this in a view class, in order to have your setter set the view as needing display.

Property keywords

The property keywords affect the responsibilities of the accessor implementations. If you synthesize the



Differentiate Yourself: Get Certified

- Mac OS X Lion 10.7 Support Essentials & Server
- Mac OS X 10.8 Support Essentials & Server
- Directory Services, Deployment, Security & Mobility
- iOS5 and Mobile Device Management
- Final Cut Pro 7 and X
- Adobe Creative Suite 5
- Logic Pro 9
- Custom Training
- 29 Years Experience



ComputerTree

Train at Our Location, or Yours!

www.ComputerTree.com

1-800-467-9820 x6



Authorized
Training Center

Back up everyone, everywhere.



CRASHPLAN™

Continuous Backup for Everyone

The easy, automatic way to protect all your data.

Individuals, businesses and enterprises around the world count on CrashPlan+, CrashPlan PRO and CrashPlan PROe to keep their data safe.

You can too! **Try it for free for 30 days.**

Peace of mind is just a few clicks away at crashplan.com



Presto! PageManager® iPad Edition

The Document Solution for Your iPad



Available on the iPhone
App Store

- Re-sizable thumbnail, multiple page preview
- Cloud connections (GoogleDoc, Dropbox, EverNote, MobileMe)
- WiFi upload/download with local computer via an internet
- Direct convert to PDF
- PDF editing tools, annotation
- Drag-&-drop files
- Import file from photo album
- Merge or split PDF
- Open with other apps
- Full page reading view
- Document Pallet

Presto! PageManager® 9

The Ultimate Document Solution



- Stack files with various formats for easy page handling
- Wide-range of PDF conversion and features including searchable PDF
- The embedded scan module (TWAIN) supports a variety of scan option
- Document pallet for easy organization
- Document inbox for monitoring incoming files
- Filter the types of documents for quick view

implementations, the keywords tell the compiler what its implementation must do. If you write them yourself, the keywords tell you what your implementation must do.

- **atomic/nonatomic:** Whether the accessors get and set the property atomically, such that the property can never be observed in a state where it has been partially changed. **atomic** is the default, but if you write the accessors yourself and don't take any special atomicity precautions, you should declare the property as **nonatomic**.
- **readwrite/readonly:** Whether the property's value can be changed. A **readwrite** declaration declares both a getter and a setter, whereas a **readonly** property declaration declares only a getter. **@synthesize** will implement only the method or methods declared by the **@property**, so if you declare as **readonly** and then **synthesize**, the compiler will implement only a getter.
- **getter=foo/setter=setFoo::** By default, the selectors of the implicitly-declared getter and setter follow the naming pattern for accessors compliant with Cocoa's Key-Value Coding. About the only reason to change the getter selector is to rename a Boolean property's getter from "foo" to "isFoo", which is also KVC-compliant and is more readable for some properties (for example, "isEnabled" for a property named "enabled"). I have never encountered a reason to change the setter selector.

strong/copy/weak/unsafe_unretained/assign: The memory management policy. More on this in a moment.

Memory management

A property can have any of five memory-management policies.

strong is a direct ownership. Assigning an object pointer to a **strong** property establishes an ownership by the property holder of the other object. This is the property equivalent to the variable keyword **__strong**.

copy is also an owning relationship, but with a twist: the property holder may become an owner not of the object you provided, but of a copy of that object. This is most useful with mutable objects such as **NSMutableStrings**, **NSMutableArrays**, etc.

The way the setter for a **copy** property works is to send a **copy** message to the object you provided; that object must respond to the message by returning either a copy of itself, or itself. The object that the **copy** message returns is the object that will become the new value of the property. (If it doesn't respond to **copy** at all, that's a bug in your program that will cause an exception.)

weak is the property equivalent to the variable keyword **__weak**: it establishes a zeroing weak reference. The property holder knows about the value, but does not own it (at least not through this property), and if the object that is the value dies, the property will be automatically set to **nil** (on Mac OS X 10.7/iOS 5 and later).



`unsafe_unretained` is the property equivalent to the variable keyword `__unsafe_unretained`: it does not establish an ownership (hence “unretained”), but if the object dies, the property will continue to point to it, leaving it possible to send a message to a dead object (hence “unsafe”). You generally should not use this keyword.

`assign` is a synonym for `unsafe_unretained`, but since it does not explicitly state how unsafe it is to use for object pointers, you should only use it for non-object values, such as numbers and structures. It's also the default for properties, which encourages you to always specify an explicit policy (usually `strong`, sometimes `weak`) for object pointers.

Properties vs. instance variables

Properties are often public (declared in the `@interface`), whereas instance variables are best kept private (declared in the `@implementation`).

Properties can only be declared in an `@interface`, whereas instance variables can be in either the `@interface` or the `@implementation`.

Properties can be added by categories, whereas instance variables cannot.

You can declare an instance variable with the same type and name as the property, but you don't need to: when you synthesize a property, you also synthesize an instance variable that matches the declaration of the property.

A synthesized instance variable has the same ownership policy as the property it came from, which means that you can

leave a property set to the default policy (`assign/__unsafe_unretained`) and synthesize it, and the default policy for instance variables (`__strong`) will not keep the object alive anyway. (That said, for properties and instance variables that refer to objects, you should use `weak` or `__weak` whenever possible.)

While a property can create an instance variable (as part of a `@synthesize` directive), an instance variable never creates a property.

A property implies accessor methods, whereas an instance variable does not.

Some notes about style

As a general rule, it is better to be explicit than implicit.

Accordingly, it's almost always better to make a property for everything, even things that you will hold privately. (You can declare the property in a class extension to keep it private.)

For one thing, as previously shown, a property declaration should include the ownership policy for the property. Explicitly declaring that isn't only a good idea for public properties; it's just as good for private properties.

For another, a property includes accessors. These provide a few advantages over an instance variable alone:

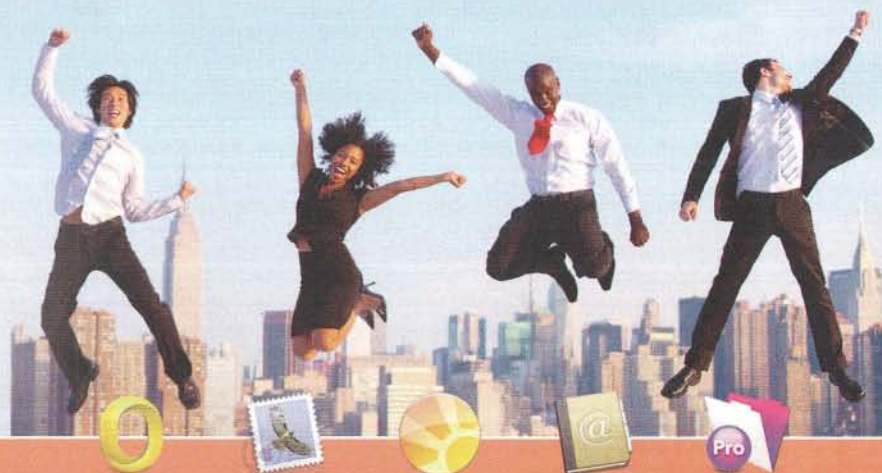
- When something uses Cocoa's Key-Value Coding system to interact with your object, KVC will call your accessors. Without accessors, it will (by default) access your instance variable directly—not something you generally want any




Direct Mail



jumpstart your
small business
with powerful email marketing



Download for **FREE** Today at
www.directmailmac.com

 Designed and built only for Mac

- Seamlessly integrates with the Mac applications you already use
- Create and send email campaigns with ease
- Real-time tracking to monitor effectiveness
- **No monthly fees**

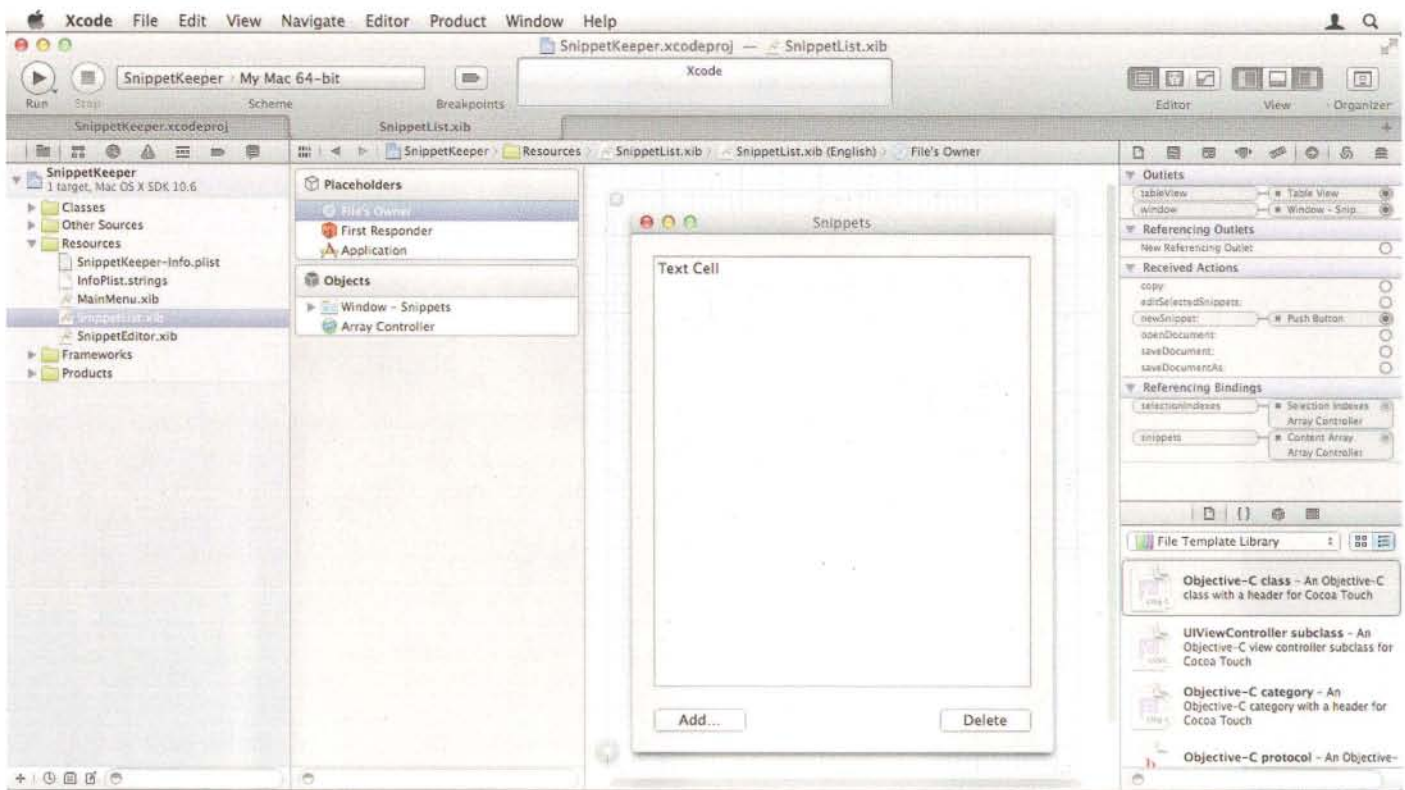


Figure 1 — Xcode's nib editor (center half), with the File's Owner selected in the objects list (middle-left quarter) and outlet properties listed in the Connections Inspector (far-right quarter).

object to be able to do.

- You can set breakpoints on either or both accessors.
- You can override either or both accessors, either to add logging statements or experimental bug-hunting code, or to implement custom behavior you'll want in the final product (e.g., sending yourself `setNeedsDisplay:` in a view).

For these reasons, you generally should make everything a property, whether you make it public or private.

Using properties in `init`

There is a risk to using your properties in `init`. A property access is a message, so if you have (or a subclass has!) a custom implementation of the accessor, using that implementation on an object you haven't fully initialized yet may be unsafe.

As an example, let's say you have a class `Foo` and a subclass `Bar`. `Foo` declares an `image` property and synthesizes it, and `Bar` overrides the accessors for the property with some custom logic. `Bar`'s implementation looks like this:

```
@interface Bar ()
@property(weak) NSCache *imageCache;
@end
```

```
@implementation Bar
```

```
@synthesize imageCache;
```

```
- (id) init {
```

```
    if ((self = [super init])) {
        self.imageCache = /*Obtain the cache from somewhere*/
    }
    return self;
}

- (NSImage *) image {
    return [self.imageCache objectForKey:self];
}

- (void) setImage:(NSImage *)newImage {
    [self.imageCache setObject:newImage forKey:self];
}

@end
```

`Foo`'s implementation looks like this:

```
@implementation Foo
```

```
@synthesize image;
```

```
- (id) init {
    if ((self = [super init])) {
        self.image = [NSImage imageNamed:NSImageNameUser];
    }
}

@end
```

The first thing `Bar`'s implementation of `init` does is call `[super init]` (as it should). `Foo`'s implementation then sends to `self` a `setImage:` message (written above as a property access expression).

A man and a woman are in a workshop. The woman is in the foreground, focused on painting a geometric pattern on a wooden block with a green brush. The man stands behind her, observing. The workshop is filled with various tools, a blender, and several wooden blocks already decorated with colorful geometric patterns. A large, colorful geometric artwork is visible in the background.

The idea: Turn surfboards into art

Serena Mitnik-Miller and Mason St. Peter of custom surfboard collective Two Birds Fly are living proof that when you're passionate about an idea, and you free it, beautiful things happen. Whether they're managing orders or creating promotional materials, working in a trusted productivity suite helps them take care of business and make sure their big idea takes flight.

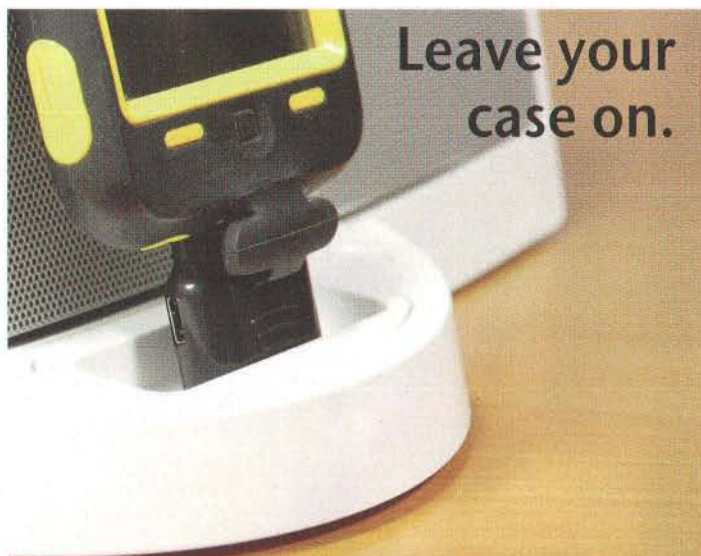
Free the ideas.



Introducing **Office:mac**²⁰¹¹

Transcend platforms and free your ideas
with Microsoft® Office for Mac 2011.

Learn more at officeformac.com/freetheideas



Leave your
case on.

dockStubz

Pass-through adapter
for iPod and iPhone.

15% off code: mt1111
(expires Dec 30)

CableJive
www.cablejive.com



Completely
redesigned
user interface
& much more.

Graphic Converter 7

Open, save & edit
almost any picture
file format

Metadata editing, batch
conversion & web
catalog generation.

www.lemkesoft.com

But remember that `self` is a `Bar` instance, and `Bar` overrides `setImage:`. Moreover, `Bar`'s implementation of `setImage:` relies on something that its `init` will set up, but hasn't yet because it's still waiting for `[super init]` to return.

In this case, what will happen is that `-[Bar setImage:]` will send that `setObject:forKey:` message to `self.imageCache`, which is still `nil`, so nothing will happen. The object that should have an image by default will not have one because of this bug. (Worse, if only some of these objects are `Bar` instances while others are direct `Foo` instances, then some—the `Foo` instances—will have images by default while others won't, and you may not be able to tell which are which in your user interface.)

The upshot of all of this is that it's safer to access instance variables directly, and not use properties, within implementations of `init` and other initializers.

On the other hand, that's also a rare situation. The above example is quite contrived. Most of the time, all relevant properties are `@synthesized` and, even if you have a subclass, it has no reason to override them.

On top of that, sometimes you *want* an accessor implementation to run. For example, perhaps the property is declared as `copy`, and the object that `init` obtains and wants to store in the property may be mutable. In that case, you would want to store a copy, not the original, mutable object, in the property. Writing a `copy` message in `init` is a thing repeated; you have specified that it must be copied both in the property declaration and in `init`. This violates the principle of DRY (Don't Repeat Yourself).

So it's a trade-off. There are reasons behind each practice.

My recommendation, especially under ARC (which eliminates 99% of memory-management code that would have had to be repeated without it), is to assign directly to the instance variable in `init` as your normal behavior, and only use an accessor—or any other method of `self`—from `init` only when you really need to. For a synthesized property declared as `copy`, I would use the property from `init`.

Accessing properties vs. instance variables

Think fast: what does this do?

```
myImage = [pictureTaker outputImage];
```

Is `myImage` a property, an instance variable, a local variable, or something else?

The answer is that it's a variable of some kind. You would have to find the declaration to be sure, but it could be either a local variable or an instance variable (or a global, though that's less likely).

`myImage` could be *declared* as a property, but remember that synthesizing a property also generates an instance variable. As long as the `@synthesize` directive for that property came before that statement, the statement would be valid, and it would use the instance variable that you synthesized for the property.

Whether there is a property by that name or not, "`myImage`" is *not* an access of a property. A property access is

Remember when typing felt good?



matiastm
tactilepro
a better keyboard for your Mac

Mac users who crave the **satisfying “click”** of Apple’s legendary *Extended* and *Extended II* keyboards will love the **Tactile Pro**. Unlike other keyboards made today, each key is built on a premium Alps mechanical keyswitch. They feel better, and you’ll type faster.

www.matias.ca

a message, and that isn’t any kind of message expression. It accesses the `myImage` variable directly.

If you wrote a custom `setMyImage:` implementation, it doesn’t get called. Even if you didn’t write a custom implementation, if you declared the property as `copy`, the copy is done by the synthesized setter method, which doesn’t get called.

The correct code would be this:
`self.myImage = [pictureTaker outputImage];`

Which translates to this:
`[self setMyImage:[pictureTaker outputImage]];`

Either of these expressions uses the property’s accessor method, so the memory-management policy—and any other desired side effects—will be upheld.

When you really do want to access an instance variable, not a property, you can be explicit about it by using the instance-variable access syntax described above:
`self->myImage = [[pictureTaker outputImage] copy];`

The “dot syntax”, `self.myImage`, accesses the property (sends a message), whereas the arrow syntax accesses the instance variable directly, explicitly. The arrow syntax does not require a property and does not use any accessor methods.

Most of the time, you don’t want to access instance variables directly because properties are safer. Whenever you do something that’s less safe, it’s good to be explicit about it.

Outlets

When you create your user interface in nibs or storyboards (I’ll refer to both collectively as “nibs” from here on), you often need to be able to refer to those objects in your code. For example, a controller may need to get a value from a view in order to set it somewhere in the model.

It’s easy to refer to an object that you create in the code, but what about objects you create in nibs?

That’s where outlets come in.

An outlet is a property that you declare with the `IBOutlet` keyword (“IB” standing for Interface Builder, another name for Xcode’s nib editor view), like so:
`@property(weak) IBOutlet NSTextField *nameField;`

When you declare that a property is an outlet, it will appear in the nib editor when you right-click on the object with the property, or when you select the object and look in the Connections Inspector. Only properties with the `IBOutlet` keyword will show up; properties that don’t have it aren’t outlets.

Once you’ve created an outlet and revealed it in the nib editor, you can drag from its socket (the circle on the right side of the outlets list) to any suitable object. For example, given the outlet above, you could drag from the `nameField` outlet to any `NSTextField` in the same nib. That connects the outlet to the `NSTextField` object. Loading the nib will set the property to that object.

SAINT[®] for Mac OS X

SAINT for Mac OS X Lion Now Available

Integrated Vulnerability Scanning, Penetration Testing,
and Checklist (Benchmark) Compliance.



Vulnerability Scanning

Assess any target with an IPv4, IPv6, or URL with pre-defined policies for PCI, HIPAA, FISMA, and more.

Identify CVE, OSVDB, IAVA, OVAL, and more.



Penetration Testing

Exploit vulnerabilities to gain remote access.

Run social engineering, phishing assessments, and more with the exploit tools suite.



Checklist Compliance

Show compliance with FDCC & USGCB security configuration policies defined by NIST.



For more information—

www.saintcorporation.com/mac
1-800-596-2006

SAINT is SCAP validated by NIST & is a certified PCI approved scanning vendor (ASV).

Don't get Lost in the Crowd...



...let an Evolis card printer Identify you!

Fully compatible with Mac®, the Pebble⁴ ID card printer is the perfect solution to print all your ID cards in high-resolution. Whether you need Employee ID's, Student ID's or Loyalty cards, the Pebble ID card printer will help you stand out from the crowd.

Call TransTech Systems today

1-888-843-3643

to get connected with your
local Evolis partner!



Pebble⁴
Color Single-sided
card printer



TransTech is an Official Evolis Solutions Provider

Tel: 1.888.843.3643 • Fax: 503.682.0166 • email: sales@ttsys.com • www.ttsys.com

Mac is a registered trademark of Apple Inc.



Outlets vs. properties and instance variables

Any property or instance variable can be an outlet. The way to make it one is to include the word `IBOutlet` before the type name in its declaration.

Prior to the introduction of property declarations, you could only use `IBOutlet` on instance variables, so only instance variables were ever outlets. Nowadays, it's best to make all of your outlets properties.

Memory management

The memory management policy you should give to an outlet property depends on the object's position in the nib.

A nib contains one or more objects, and some of these objects—particularly views, windows, and (on the Mac) menus—may contain other objects. Objects that are contained by the nib directly, not by another object in the nib, are called top-level objects.

The File's Owner should, as its name says, own all of the top-level objects, so the outlets to those objects should be *strong*. The File's Owner doesn't need to own any objects that are and always will be contained within a top-level object; those will remain alive anyway, so it makes clean-up simpler to declare those properties as *weak*.

There are some extra wrinkles on the Mac. For details, see the “Nib Files” chapter of the Resource Programming Guide in Apple's documentation.

Public or private?

You used to have to declare outlets in the header, but with recent versions of the Clang compiler, you can declare them in a class extension in the module file instead. That will look something like this:

```
@interface MyWindowController ()
@property(weak) IBOutlet NSTextField *ageField;
@end
```

There's a common rule in object-oriented programming called the Law of Demeter, which states that every object should only talk to “itself or its friends”, the latter part meaning any objects it creates or has passed to it by something else. An object shouldn't talk to another object's friends.

From that, it follows that you generally shouldn't make an outlet public. For example, if the `ageField` property shown above were declared in that window controller's interface, other objects could get or set the field's value directly.

That may sound fine; why make the path indirect? The reason is because then your entire program becomes entangled with the UI. You can't change the UI without having to change things all over your code base.

audioengine

Join the computer audio revolution

Free Shipping
Buy Online Direct
audioengineusa.com



A5+ shown in
Hi-Gloss White

audioengine 5+
www.audioengineusa.com

Premium Powered Speaker System

Built-in power amplifiers
Remote control
USB power/charge port
High-quality connectors
Hand-built cabinets
Starting at \$399/pair

Audioengine delivers sound and features at prices that set the standard for affordable high-quality audio. Connect your iDevice, computer, TV, or any audio component for great stereo sound.

Cables included • 30-day audition • 3-year warranty

Visit our website for more info, reviews, and awards: audioengineusa.com



SHATTERED MEMORIES?



WE CAN HELP

Recover Your Lost Files Now!

We Provide Data Recovery Software And Services For All Types Of Media



www.LC-Tech.com/mt

A GLOBAL LEADER IN DATA RECOVERY





Wi-Fi^licious

Simply Smarter Wi-Fi
in the Age of the Mobile Internet.



As an example, what if you wanted to add a slider or a stepper alongside the field? Then you would have to change every class that sets the value in the field to also set it in the slider or stepper—and, when you need to get the value, how would you tell which of those controls the user had changed last?

It's better to give the window controller a public property for the value, and keep its outlets private. The window controller, and the window controller alone, is responsible for communicating that value back and forth with the UI. Whatever object created the window controller will get the value from it, and pass it along, if necessary, to wherever it is ultimately needed (ideally in a model object).

Objects in multiple nibs

There is no such thing.

An object cannot be in two nibs at once. A nib is an archive of objects; when you load the nib, you are unarchiving (deserializing) those objects. There's no way for the nib loader to know that an object in one nib is meant to be the same object you got from another nib, so that can't happen: they are always two separate objects.

If you connect the same outlet in two different nibs to two different objects, then, regardless of how similar you make those objects or the fact that you connected the same outlet to both of them, they remain different objects, and one of them will replace the other in your outlet.

This is a common cause of confusion: you may load one nib, perform some setup in code on the object you got from it, and then load another nib, and wonder what happened to the changes you made in code. What happened was that you loaded another object that had never had those changes made to it, and because you had hooked it up to the same outlet, it replaced the object that you previously loaded from the first nib.

So remember that all of the objects within a nib exist *only* within that nib; you cannot share an object between multiple nibs.

Wrapping up

You've learned how to connect objects, both those you've created in code and those you've created in nibs, to each other. You've learned about the difference between properties and instance variables, how one uses the other, and what it means to make an outlet.

Next month, we'll (finally) look at the frameworks! Cocoa, Cocoa Touch, the differences between them, Core Foundation, Core Graphics, Core Animation, what's left of Carbon, and more.



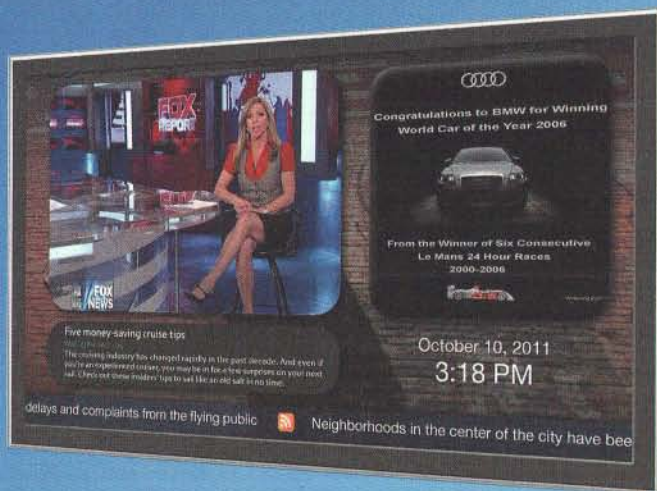
About The Author

Peter Hosey is a programmer of numerous open-source Mac applications. He answers programming questions on Stack Overflow and can be reached via Twitter as @boredzo.



MEDIASIGNPRO

Creating Digital Signage Has Never Been Easier!



Faster. Smarter. Better.

Software Exclusively for Mac
Compatible with Lion OS
All-in-One Solution
Creator, Scheduler and Player
No Monthly Fees

Fully Customizable Layouts
Social Network Capabilities
Remote Management
No Rendering Required
No Server Required



Affordable and Fully Customizable Digital Signage Software

Free Trial Available

Receive a 20% discount by using the promotional code - MacTech20

Download our free White Papers and learn more about how to use digital signage.



MEDIASIGNPRO

sales@mediasignpro.com

mediasignpro.com

866-991-3713



An Introduction to GNU Octave

Is Octave a viable MATLAB alternative?

by Mihalis Tsoukalos

Introduction

This article is an introduction to GNU Octave, a high-level interactive language, primarily intended for numerical computations that is compatible with MATLAB.

While GNU Octave is a powerful software package, it is free. Octave is often considered a MATLAB alternative. MATLAB, a product of MathWorks, is not free. Although Octave tries to be as compatible as possible with MATLAB, there are infrequent cases that you cannot directly execute Octave code in MATLAB and vice-versa.

What is Octave?

The name Octave comes from the chemist Octave Levenspiel. GNU Octave, an official GNU project, is a project that started by James Rawlings and John Ekerdt but its main developer is John Eaton.

It can solve many different problems using its native functionalities but it can also be extended using its complete programming language. Its programming language works like an interpreter—the code is executed line by line every time you run an Octave program. Octave also has plotting capabilities that are covered in this article.

Because Octave is mainly for performing mathematical and numerical computations, it is not a replacement for general-purpose programming languages like C, Objective-C or C++.

Installing and Using Octave

When I started writing this article, I had version 3.4.2 of Octave, installed using MacPorts, running on my iMac with Mac OS X 10.6.8. I was using the latest MacPorts version (the *development* version) of Octave. I installed it using the following command:

```
$ sudo port install octave-devel
```

During the writing of this article, I switched to Mac OS X 10.7.1 using the clean installation option. The aforementioned *octave-devel* version does not currently (at the time of writing this) compile on Lion, so the rest of the article uses the 3.2.4 Octave version that can be installed using the following command:

```
$ sudo port install octave
```

In order to run Octave, you just have to type “*octave*” at the UNIX shell. The output will be similar to the following:

```
$ octave
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying
conditions.
There is ABSOLUTELY NO WARRANTY; not even for
MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type
'warranty'.
```

```
Octave was configured for "i386-apple-darwin11.1.0".
```

```
Additional information about Octave is available at
http://www.octave.org.
```

```
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-
wanted.html
```

```
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a
helpful report).
```

```
For information about changes from previous versions, type
'news'.
```

```
octave:1>
```

The octave command prompt is pretty simple and does not offer much help. Type *help* for more information:

```
octave:1> help
```

```
For help with individual commands and functions type
```

```
help NAME
```


STILL SOLID. WAY COOLER.



Keep your computers indestructible,
free up crucial IT resources, and reduce
support costs by 63%.

Faronics Deep Freeze automatically restores
workstation configurations with every reboot.

For more information, visit
www.faronics.com



Available for:



Lost Data?



We can save it!

Fast • Reliable • Certified
Secure • Data Recovery

800.440.1904

www.drivesavers.com

©2010 DriveSavers Data Recovery Inc.

BUNDLED PHONE & INTERNET SERVICE

FROM \$459 FLAT RATE

Dynamic Allocation T-1
Up to 16 Business Lines
Unlimited Local Service
Unlimited Site to Site Calling
2,000 Minutes of Long Distance
or Toll Free

Voice Mail, Call Forwarding, 3-Way Calling, Call Hold,
Pickup and Transfer, Call Waiting, Last Number
Redial, DID, and DOD, Caller ID and more!

www.lowcostdialing.com

800-906-8686

(replace NAME with the name of the command or function you would like to learn more about).

For a more detailed introduction to GNU Octave, please consult the manual. To read the manual from the prompt type

```
doc
```

GNU Octave is supported and developed by its user community.

For more information visit <http://www.octave.org>.

```
octave:2>
```

You can create a variable called "mactech" and print its value as follows:

```
octave:2> mactech=1
```

```
mactech = 1
```

```
octave:3> mactech
```

```
mactech = 1
```

```
octave:4> MacTech
```

```
error: 'MacTech' undefined near line 4 column 1
```

The last error message shows that Octave is case sensitive, as it cannot recognize a variable named "MacTech".

Octave also supports complex numbers. If you are good at Mathematics you should remember that complex numbers have a real part and an imaginary part and can be written in the $a + bi$ format, where i is the square root of -1 . The following example shows how you can use Octave to work with complex numbers:

```
octave:4> c1 = 1 + 2i
```

```
c1 = 1 + 2i
```

```
octave:5> c2 = -1 + 2i
```

```
c2 = -1 + 2i
```

```
octave:6> c3 = 4 - 5i
```

```
c3 = 4 - 5i
```

```
octave:7> c4 = c1 + c2
```

```
c4 = 0 + 4i
```

```
octave:8> c5 = c3 * c2
```

```
c5 = 6 + 13i
```

```
octave:9> real(c5)
```

```
ans = 6
```

```
octave:10> imag(c4)
```

```
ans = 4
```

If you want to see the existing variables, you should execute the "whos" command:

```
octave:12> whos
```

Variables in the current scope:

Attr	Name	Size	Bytes	Class
====	=====	=====	=====	=====
	ans	1x1	8	double
	c1	1x1	16	double
	c2	1x1	16	double
	c3	1x1	16	double
	c4	1x1	16	double
	c5	1x1	16	double
	c6	1x1	16	double
	mactech	1x1	8	double

Total is 8 elements using 112 bytes

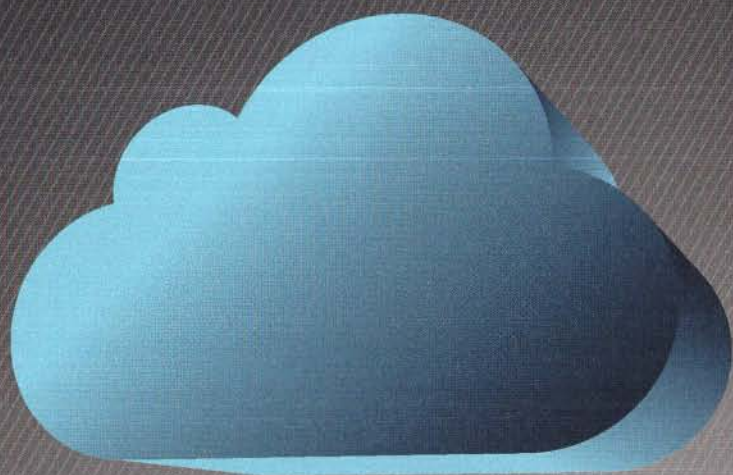
In order to delete a variable, you should use the "clear <variable name>" command. Be very careful with the "clear" command as it works with the wildcard *. For example, in



Dev Cloud

Acquia's cloud platform, now sized for developers

The same architecture and toolset that serves billions of pageviews monthly for sites like Warner Music, Stanford University and Al Jazeera, is now available in a single-server option, for the **ultimate Drupal developer experience.**



Acquia™

acquia.com | 888-922-7842

@acquia | @acquiacommons | @drupalgardens

order to delete all the complex variables that start with *c*, you should run the following command:

```
octave:13> clear c*
octave:14> whos
Variables in the current scope:
```

Attr	Name	Size	Bytes
Class	====	====	=====
double	ans	1x1	8
double	mactech	1x1	8

Total is 2 elements using 16 bytes

If you want to delete **every** variable, just type “*clear*”.

You can also create matrices, vectors and text variables in Octave.

Octave Functions

Octave allows you to create your own functions, which is very practical when you want to execute the same commands many times. You can even interact with the user asking her for input. In this part, a user-defined function called *my_function* is programmed.

The function code, saved in a file named *my_function.m*, is the following:

```
#
# Usage:
# [] = my_function
#
# Returns YES is the given number is a prime number
# NO otherwise
#
# Programmer: Mihalis Tsoukalos
# Date: Friday 09 September 2011
#

function [] = my_function

    n = input("Enter an Integer number: ");

if ( ischar(n) )
    usage("Input must be an Integer number");
endif

    NO = 0;
    min = 2;
    max = floor(n/2);

    for y=min:1:max
        if ( rem(n,y) == 0 )
            disp("NO");
            NO = 1;
            break;
        endif
    endfor

    if ( NO == 0 )
        disp("YES");
    endif
endfunction
```

What this simple function does is to check whether the given number is a prime number. A prime number is an **Integer** number that can only be divided by 1 and itself. The

algorithm checks the remainder of the input against a range of numbers. If the remainder is 0 for one of the numbers, then the given Integer is not a prime number. You can use the *my_function* as follows:

```
octave:6> my_function
Enter an Integer number: 101
YES
octave:7> my_function
Enter an Integer number: 111
NO
octave:8> my_function
Enter an Integer number: "12"
usage: Input must be an Integer number
error: /Users/mtsouk/docs/article/Octave.MT/my_function.m
at line 17, column 2
octave:9>
```

You do not have to manually load the file inside Octave. All you have to do is to be inside the directory where the file is. Also, if you execute “*help my_function*” you will get the following output:

```
octave:1> help my_function
'my_function' is a function from the file
/Users/mtsouk/docs/article/Octave.MT/my_function.m
```

```
Usage:
[] = my_function
```

```
Returns YES is the given number is a prime number
NO otherwise
```

```
Programmer: Mihalis Tsoukalos
Date: Friday 09 September 2011
```

Additional help for built-in functions and operators is available in the on-line version of the manual. Use the command “*doc <topic>*” to search the manual index.

Help and information about Octave is also available on the WWW at <http://www.octave.org> and via the help@octave.org mailing list.

Does the output remind you of something? Yes, the comments at the beginning of the *my_function.m* file are displayed on screen!

Octave Packages

GNU Octave also supports packages programmed by others that extend its capabilities. MacPorts offers a big list of Octave packages and if you want to use some of them it is better to install the MacPorts version if available.

If you want to see the list of the currently installed packages, you should type the following Octave command:

```
octave:1> pkg list
Package Name | Version | Installation directory
+-----+
general *| 1.1.3 | /opt/local/share/octave/packages/general-1.1.3
plot *| 1.0.7 | /opt/local/share/octave/packages/plot-1.0.7
octave:2>
```




real.studioTM

We're just easier.

Targeting multiple platforms shouldn't be so hard. Use Real Studio to create anything from simple utilities to powerful enterprise-class applications. It's the only object-oriented, cross-platform development tool that lets users at all levels quickly build native applications for Mac OS X, Windows, Linux and the web. **Work smarter with Real Studio.**

www.realsoftware.com/realstudio


```
Package name:
  general
Version:
  1.1.3
Short description:
  General tools for octave.
Status:
  Loaded
octave:3>
```

```
Package name:
  benchmark
Version:
  1.1.1
Short description:
  The package contains code used to benchmark speed of
  Octave.
Status:
  Loaded
octave:6> pkg unload benchmark
octave:7> pkg describe benchmark
--
Package name:
  benchmark
Version:
```

```

    The package contains code used to benchmark speed of
Octave.
Status:
    Not loaded
octave:8> pkg load benchmark

```

(In Octave, a polynomial is represented by a vector. To create a polynomial, you enter each coefficient of the polynomial into the vector in descending order.)

iDeveloper.tv



Start a Web Site Today!

from only **\$4.95** /month

Get first month for a penny!
Coupon code: **MACTECH**



Powered by 130% Texas Wind Energy!

100% Mac-compatible Web Hosting



- 99.9% Uptime
- 45-day Money Back Guarantee
- Over 4,500 Free Web Design Templates
- Free Site Builder Software
- In-house Support Available 24x7x365

Sign up at www.HostGator.com/mactech
and get the first month for just a penny!

Already have a web site?
We'll transfer it for **free!**

www.HostGator.com

866.96.GATOR

In order to evaluate the polynomial for a given value (e.g. $x=3$), you should use the *polyval* (type “*help polyval*” inside Octave for more information) function as follows:

```
octave:5> polyval(d, 3)
ans = 32
octave:6>
```

Using the *roots* command, you can find the roots of a polynomial. The following example calculates the roots of both *c* and *d* polynomials:

```
octave:7> roots(c)
ans =
```

```
-5.37228
0.37228
```

```
octave:8> roots(d)
ans =
```

```
-1.70998 + 0.00000i
0.85499 + 1.48088i
0.85499 - 1.48088i
```

It is visible that the *c* polynomial has two roots that are both real number whereas the *d* polynomial has three roots—two of them being complex numbers.

Octave can also calculate the indefinite integral (using the **polyint()** function) and the derivative (using the **polyderiv()** function) of a polynomial. The following commands show how:

```
octave:12> polyint(c)
ans =
```

```
0.33333 2.50000 -2.00000 0.00000
```

```
octave:13> polyint(d)
ans =
```

```
-0.25000 0.00000 0.00000 5.00000 0.00000
```

```
octave:14> polyderiv(c)
ans =
```

```
2 5
```

```
octave:15> polyderiv(d)
ans =
```

```
3 0 0
```

The *polyderiv(polyderiv(d))* command—not shown here—calculates the derivative of the derivative of the *d* function.

Plotting with Octave

First, let's try a simple plotting command that creates an impressive output that can be shown in figure 1. The command, that is included in every Octave installation, is the following:

```
octave:1> surf(peaks)
```

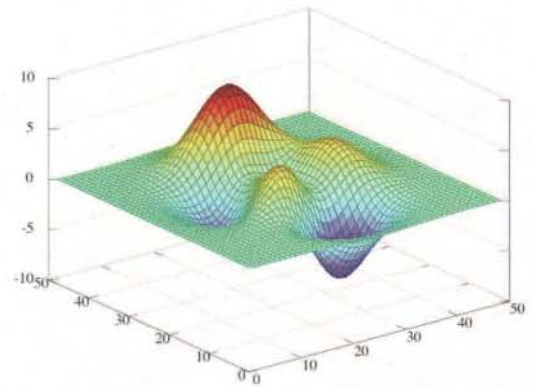


Figure 1: The output of the *surf(peaks)* command

Note that in order to see the output of the command you should have the X11 environment installed on your Mac OS X system. The X11 environment ran automatically on my Lion system after executing the Octave command.

It will now be explained how to plot functions, save and customize their output. The first command creates a new polynomial, named *plotME*:

```
octave:34> plotME = [1 -2 3 -1]
plotME =
```

```
1 -2 3 -1
```

It would seem logical to try to plot it at once, using the “*plot*” command that you suspect that it exists. If you execute “*plot(plotME)*”, you will get figure 2.

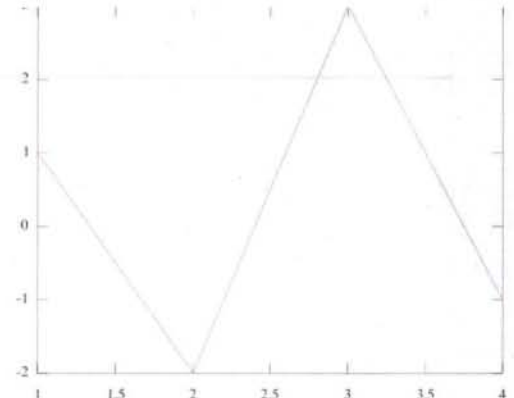


Figure 2: *plot(plotME)*

Figure 2 does look neither good nor professional. The problem with it is that it calculated the polynomial's values for 1, 2, 3 and 4 and then connected the points. This is not what we wanted. We want more accuracy, which means that more points have to be connected when drawing the graph. Also, we want to be able to define the limits of the x axis ourselves.

The Octave way to do that is to define a new variable, a vector, that contains the x axis limits as well as the step value. If we want to draw the *plotME* function in the $[-3, 3]$ space using a .01 step, then we should define a vector variable as follows:

```
octave:37> limValues = [-3:0.01:3];
```


Note: If you forget to put the semicolon at the end of the command, you will get lots of useless output. The last step is to connect the `limValues` vector with the `plotME` function as follows and then call the `"plot"` function:

```
octave:39> newPlot = polyval(plotME, limValues);
octave:40> plot(limValues, newPlot)
```

The first command creates a new variable called `newPlot` that contains the output values. Then, every value pair is plotted using the `plot` command. The output can be seen in Figure 3.

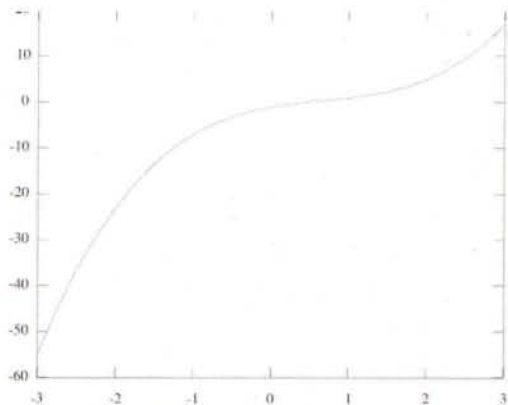


Figure 3: The `plot(limValues, newPlot)` command output

Although the output is much better than before, I think that we should try to enhance it a little. The following commands finally create figure 4:

```
octave:43> plot(limValues, newPlot, "linewidth", 3);
octave:45> set(gca, "ylabel", text("string", "f(x)", "fontsize",
25))
octave:49> grid on
octave: 50> legend("f(x)")
octave:51> print("figure4.png", "-dpng");
```

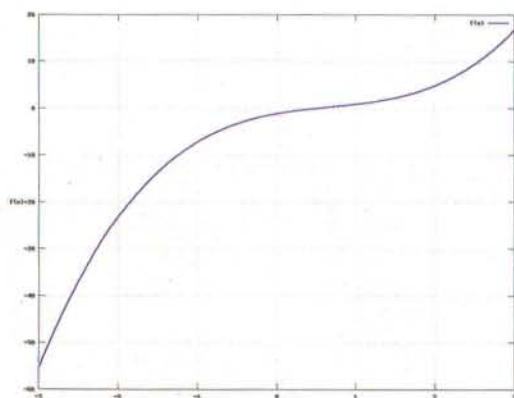


Figure 4: Beautifying the plot command output

The last command saves the output graphics in png format using the `"figure4.png"` filename. The other supported formats are Encapsulated Postscript (eps), Postscript (ps), PDF

(pdf), jpeg (jpg or jpeg), GIF (gif), TeX picture (tex) and LaTeX picture (pslatex).

Summary

Octave is a very powerful tool for numerical and mathematical computations and if you need such a tool, you should definitely consider it. I think that you should by now wonder why Octave is free; the answer is GNU.

GNU Octave is a viable MATLAB alternative!

Bibliography & Web Links

Octave: <http://www.gnu.org/software/octave/>

MATLAB: <http://www.mathworks.com/products/matlab/>

Octave-Forge: <http://octave.sourceforge.net/>

M

About The Author

Mihalis Tsoukalos enjoys digital photography, writing articles and programming his iPhone 4 and iPad. He is the author of Programming Dashboard Widgets, an eBook. You can reach him at tsoukalos@sch.gr.

BREAKING NEWS

DATAJACK, LOWEST COST NATIONWIDE MOBILE BROADBAND

DATAJACK™
Take Your Internet With You



Mobile Hot Spot

Monthly Data Plans starting at **\$9.99**

Internet on the go, with DataJack's Hot Spot

MiFi 2200 Mobile Hotspot

Connect up to 5 WiFi devices at lightning fast 3G speeds. Mobile broadband on the go.



Broadband USB Modem

Easy auto-install Plug -N- Play USB device providing 3G mobile broadband access on the go.



Quamtel, Inc.
Symbol: QUMI

www.DataJack.com

Leaves of Data

Presenting data trees with AppleScriptObjC

by José R.C. Cruz

Introduction

Sometimes, data sets come in the form of a tree. These sets have relations that are too complex to be viewed with a table view. To display them, we need the aid of an outline view.

Today's article looks at how we can use the `NSOutlineView` class to present data trees. We will learn what makes a typical data tree and how to parse the items in that tree. We will examine how to use data sourcing to shuttle data to the view. And we will do all this using AppleScript and the AppleScriptObjC (ASOC) code bridge.

Once again, readers are expected to have a working knowledge of Xcode and AppleScript. The demo project itself is available from the MacTech source code repository at [ftp://ftp.mactech.com](http://ftp.mactech.com).

The Data Tree

In a typical data tree (Figure 1), items are arranged into *branches* and *leaves*. Each item forms a node that either links to other items or stands alone.

All branches and leaves share a common node known as the *root* of the tree (marked here in red). Nodes that link to two or more nodes serve as branches. Those linked to exactly one node are the leaves. So, by this definition, item 'anthropoda' (in olive green) is a branch. It attaches to the leaf node 'scorpion' (in pink) and to the second branch 'hexapoda' (olive green). Item 'anthropoda' is also attached to the root.

Now it is possible for a data tree to have only a root node and at least one leaf. Most data trees, however, must have one or more branches, or none at all.

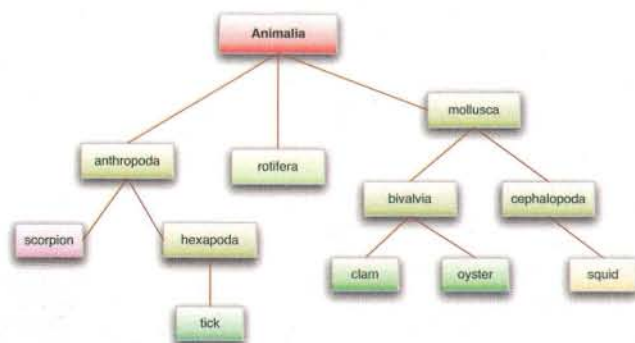


Figure 1. A typical data tree.

The number of branches separating a node from the root gives the *level* for that node. Thus in our example, the leaf item 'squid' (marked in orange) has a level of two. The leaf 'scorpion' has a level of one, the leaf 'rotifera' (lime green) a level of zero.

The greatest level sets the *depth* (or *height*) of the whole tree. Our tree, for instance, has a depth of two. Notice that the levels of all the leaves never exceed the depth of the tree.

Relations in the tree

Items in the data tree relate to each another in a unique way. If one item leads to another item, the former is considered a *parent*, the latter its *child*. Items with the same parent and level are considered *siblings*. Those with the same level but different parents are *cousins*.

A leaf item always has a branch or a root as its parent. But it cannot have any children. When a leaf item gets a child, it ceases to be a leaf and turns into a branch. A branch can have the root or a branch as its parent. And it can have a leaf or another branch as its child. Furthermore, a branch can have two or more children present at the same time.

In our sample tree, item 'anthropoda' has 'scorpion' and 'hexapoda' as its children. Both 'scorpion' and 'hexapoda' are siblings even though one is a leaf, the other a branch. The parent for item 'anthropoda' is the root 'Animalia'. Items 'hexapoda' and 'bivalvia' are cousins since they have the same level and different parents.

Uses for a tree

The data tree structure plays an essential role in many a computing process. Some files, for instance, organize data in the form of a tree. Consider the web page, for instance, which uses HTML as its format. The file starts with an `<html>` tag, marking the root node. Following the root are the `<head>` and `<body>` tags. The former holds the page's *metadata*, the latter the *main text*. Both are level-zero tags and both form branches adjacent to the root. Within the `<meta>` and `<body>` tags are other tags, marking off specific data items. Some of these tags form new branches; others form leaves.

Filesystems also make heavy use of data trees. In this case, the *physical drive* serves as the root of the tree. Off that root are the *logical volumes*, dividing the drive space into manageable chunks. Each volume then divides into *directories* and *files*. Inside each

directory are more directories or files. Both directories and volumes are the branches; the files are the leaves.

The Outline View

Due to their unique structure, data trees cannot be presented as a table. Doing so will not preserve the relations of branches and leaves. To properly display a data tree, we need the aid of an outline view.

At first glance, outline views have many of the same traits as table views. They, too, divide data into *rows and columns*. Each column gets its own header, which, when clicked, selects or sorts the column's row items. Selecting a row item extends the selection to other items on the same row.

But where the outline view stands unique is in how its *leftmost column* behaves. Leaf items appear as *normal rows* under that column. Branch items, on the other hand, appear with a *disclosure icon* next to itself. Clicking this icon starts an *expand event*, revealing the children of that branch. Clicking the icon again starts a *collapse event*, thus hiding those children. As a rule, the disclosure icon appears only when a branch has at least one valid child.

The view class

On MacOS X, the outline view is supplied by the Cocoa class `NSOutlineView` (Figure 2). This class derives from `NSTableView`, thus inheriting many of the latter's properties and methods. To add an instance of `NSOutlineView` to a xib bundle, we use the **Outline View** icon from the **Library** palette of Interface Builder.

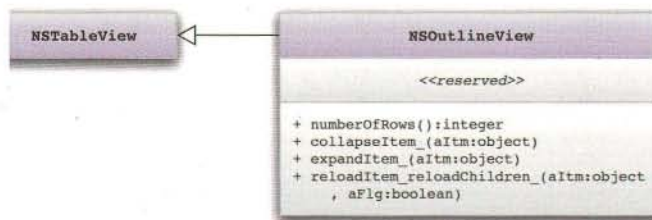


Figure 2. The `NSOutlineView` class.

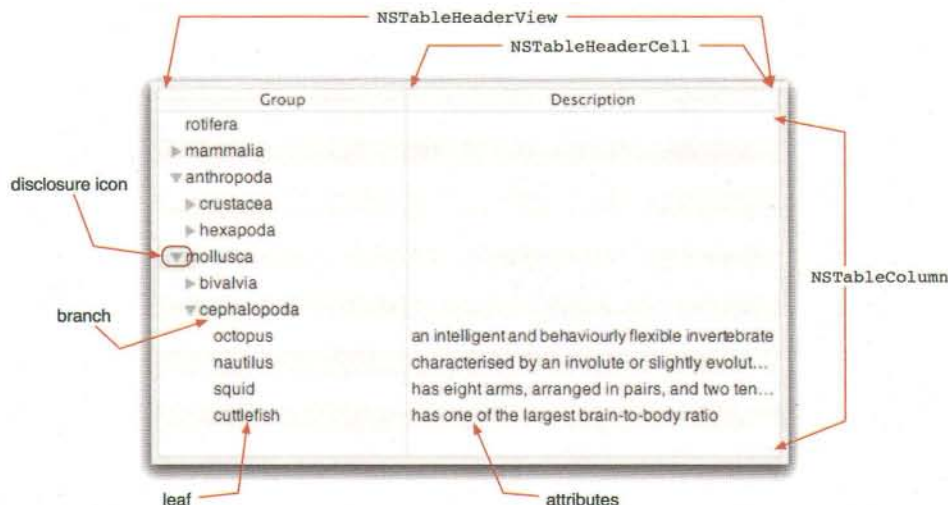


Figure 3. An instance of `NSOutlineView`.

The above class diagram shows four common methods of the `NSOutlineView` class. The method `numberOfRows()` returns the total number of row entries displayed by the view. The total includes all child items visible on the view. But as with `NSTableView`, this same total is not the number of entries *visible* to the user.

The method `expandItem_()` takes a branch item as input. It then displays the child items linked to that branch. Each child item appears as *indented rows* beneath the entry for the branch item. The method `collapseItem_()` does the opposite, removing the indented rows beneath the branch item's entry. Both methods work only if the branch item is a valid parent and its row entry has a disclosure icon.

The method `reloadItem_reloadChildren_()` takes two arguments: the tree item and a Boolean flag. If the flag is a *true* and the item is a *branch*, the method updates the row entries for that branch and its children. The update is *recursive*, affecting any child items that are valid branches themselves. Now if the flag is a *false*, only the tree item is updated.

The view layout

Figure 3 shows the parts that make up the `NSOutlineView`. On top of the view is an instance of `NSTableViewHeader`. It holds instances of `NSTableHeaderCell`, which are used to label and control each outline column. Each column is an instance of `NSTableColumn`, which uses an array of `NSCells` to display the tree data.

As expected, the leftmost column defaults to showing the branches and leaves of the data tree. A triangle marks those branches with valid children. The branch is in an expanded state when its icon points downwards. It is in a collapsed state when the icon points to the left. In short, the triangle serves the role of a disclosure icon.

The rest of the columns on `NSOutlineView` are used to show data relevant to each tree item. In the example shown, the second column describes each leaf item found in each branch.

The view behavior

Apart from its collapse and expand actions, the `NSOutlineView` class exhibits many of the same behaviors as its `NSTableView` parent. For instance, clicking a column header still causes a sort event across that column. It also selects the column for other processes. Dragging the header's borders resizes the column underneath. Dragging the column header itself repositions the column. If the dragged column is the leftmost one, that column still preserves the tree items it has displayed.

Clicking a row item also causes a selection event that extends to all rows, visible or not. A double-click causes an edit event, which can start an inline edit session or display a separate edit window.

Finally, the `NSOutlineView` supplies the same hooks for custom sort and filter routines. But extra care is required if these routines are to process the tree items properly.

Showing The Tree

To help us learn how `NSOutlineView` class works with a data tree, we will use the demo project **Gaiyo**, which meant "outline" in Japanese-Romanji. The project shows a single window view (Figure 4) on which sits our outline view. The edges of the outline view are locked with the window's edges. Resizing the window then causes the outline view to resize as well.



Figure 4. Main window layout of project Gaiyo.

The left column of the outline view gets the header label "Group", while the right gets the label "Description". Similarly, the identifier for the left column is set to the string 'group', for the right the string 'desc'. Both identifiers are set by choosing **Attributes Inspector** from Interface Builder's **Tools** menu. The data source for the entire view is served by the script object `GaiyoModel`.

The data tree itself is held by the property-list file `TreeDict.plist` (Listing 1). The tree starts with five animal phyla, three of which are shown here. Each phylum branches into two subphyla, each subphylum into member organisms. And each member organism gets a brief description of itself. In short, our data tree is a condensed subset of the entire animal kingdom.

Note that in this sample tree, the phylum `rotifera` does not contain any child entries.

Listing 1. `TreeDict.plist`—A sample data tree.

```
<dict>
  <key>rotifera</key>
  <dict/>
  <key>arthropoda</key>
  <dict>
    <key>crustacea</key>
    <dict>
      <key>barnacle</key>
      <string>exclusively marine, and tends to live
in...</string>
      <key>crab</key>
      <string>has a reduced abdomen entirely hidden
by...</string>
      <key>shrimp</key>
      <string>found globally in both salt and fresh
waters</string>
      <key>lobster</key>
      <string>economically important as seafood</string>
    </dict>
    <key>hexapoda</key>
    <dict>
      <key>grasshopper</key>
      <string>also known as a short-haired
grasshopper</string>
      <key>bed bug</key>
      <string>parasitic insects that prefer to
feed...</string>
      <key>tick</key>
      <string>ectoparasites that subsides on
blood...</string>
    </dict>
  </dict>
  .
  .
  .
</dict>
```

Again, the complete Gaiyo project is available from the MacTech source code repository at [ftp://ftp.mactech.com](http://ftp.mactech.com).

The data source protocol

The `NSOutlineView` class provides its own protocol for moving data between the outline view and its model. In this protocol, `NSOutlineViewDataSource`, are twelve method interfaces, four of which are *mandatory* (Figure 5). All protocol methods supply the target view requesting for data. Some supply the row entry under which other tree items may be placed. If the supplied entry is a missing value, the desired items are those adjacent to the root node.

The first method `outlineView_numberOfChildrenOfItem_()` states the number of children that belong to the given entry. It returns the number as an *unsigned integer*. Note the method passes the row entry as a *generic object*. Use the `NSObject` methods `isKindOfClass_()` and `class()` to check the entry's correct type.

An Awesome Email Marketing Tool for **MACTECH READERS**

Benchmark Email is the standard in permission-based email marketing

- ✓ List Management
- ✓ Free Newsletter Archive
- ✓ Easy to Use Drag-n-Drop Email Editor
- ✓ Powerful Personalization
- ✓ Image Gallery
- ✓ Visual Graphs for Open & Clickthrough Tracking
- ✓ Spam & Spell Checkers
- ✓ Creative & Compelling HTML Templates
- ✓ Upload Your own Template
- ✓ Easy Video Integration



MACTECH users
get a 10%
Lifetime Discount
with this promo code
123923

Most email marketing services charge more money for less product. Not us. Benchmark Email's sophisticated suite of email marketing features lets you grow your list, send campaigns, track your data and even take online polls for an affordable price.

Plans starting at only \$9.99 per month

Sign up for a FREE 30 Day Trial Today!

www.BenchmarkEmail.com



800.430.4095

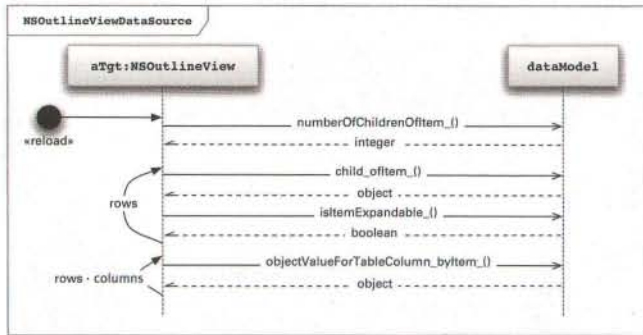


Figure 5. NSOutlineView and its data-source protocol.

The next method `outlineView_isItemExpandable_()` toggles the disclosure icon for each row entry. It returns a Boolean value. A value of true means the given entry has valid child items.

The method `outlineView_child_ofItem_()` displays the child items, if any, for each row entry. It adds the extra argument `aIdx`, which gives the row index for each child. It then returns the item as a *generic object*. The item appears only on the left-most column, which was reserved for tree items.

The last method `outlineView_objectValueForTableColumn_byItem_()` updates the columns on the outline view, including the column meant to display the tree items. It adds the argument `aCol`, which is an instance of `NSTableColumn`.

The `aCol` argument points to the column that will receive the data item. The method also returns its data item as a generic object.

Figure 6 describes how the protocol works when the outline view appears for the first time. It is when the view displays those tree items *adjacent* to the root node. On the left of the sequence diagram is the outline view `aTgt`, on the right the data model, which will be `GaiyoModel`. For brevity's sake, the method names appear without the opening term `outlineView_`.

First, `aTgt` invokes `numberOfChildrenOfItem_()`, passing along a missing value as the row entry. Then it invokes `child_ofItem_()` and displays the returned result. Next, `aTgt` invokes `isItemExpandable_()` to find out if the displayed entry needs a disclosure icon. It repeats these two steps until all the child items are displayed. Finally, `aTgt` invokes `objectValueForTableColumn_byItem_()`. It continues to call this method for all visible rows and columns. So, if the view has ten visible rows and two visible columns, the method `objectValueForTableColumn_byItem_()` will be called twenty times.

Figure 7 shows the protocol sequence when the outline view refreshes its display. This happens when the view becomes active, ready to accept user actions. In this case, the view invokes only `objectValueForTableColumn_byItem_()` for each visible row and column.

Figure 8 shows the protocol again, now responding to an expand event from the outline view. First, the view invokes `numberOfChildrenOfItem_()`, passing along the branch item to



@ScreenFlow is simply epic.
Best Mac recording software ever.



Record. Edit. Share

ScreenFlow is powerful, easy-to-use screencasting software for the Mac. With ScreenFlow you can record the contents of your entire monitor while also capturing your video camera, microphone and your computer audio. The easy-to-use editing interface lets you creatively edit your video, add additional images, text, music and transitions for a truly professional-looking video. The finished result is a QuickTime or Windows Media movie, ready for publishing to your Web site, blog or directly to YouTube or Vimeo.

Get a **free trial** at www.telestream.net/screenflow



Telestream.net

ScreenFlow 3
now available!


```
if (you_have_a_website == true) {  
  
    measure_roi = easy;  
    contact_visitors = yes;  
    real_time = of_course;  
    try_visistat = free;  
    setup = no_brainer;  
  
}  
else {  
  
    no_clue = true;  
    i_use_google = sorry;  
  
}  
  
//REAL-TIME WEBSITE TRACKING  
goto = www.visistat.com;
```



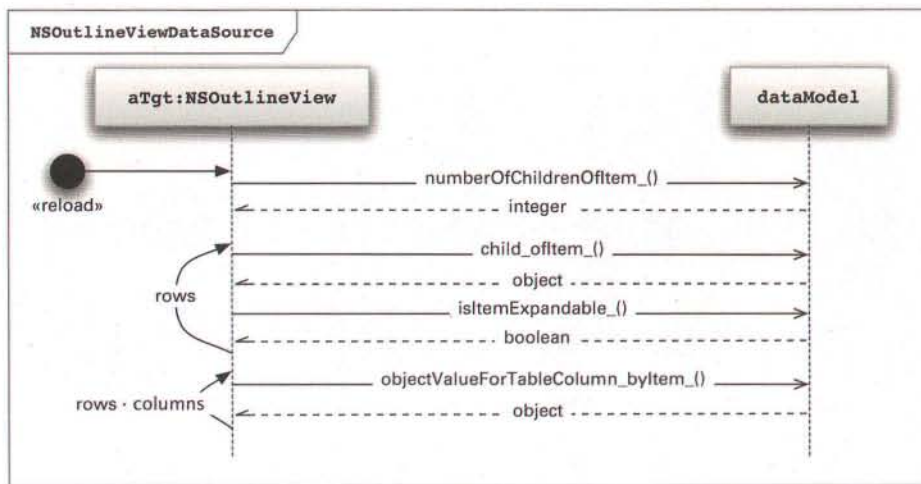


Figure 6. Displaying the root entries.

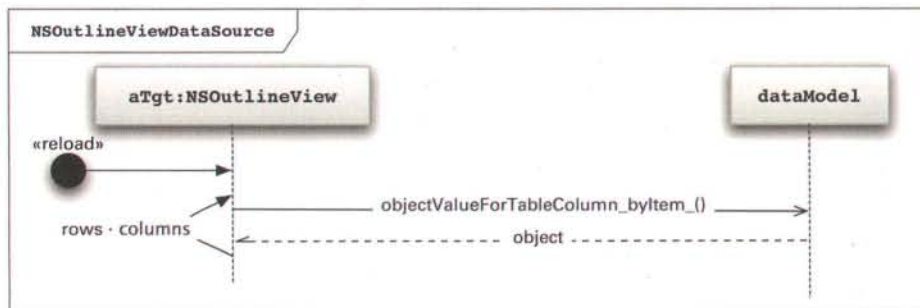


Figure 7. Updating the visible entries.

the method. It then calls `child_ofItem_()` and inserts the child item below the branch item. Next, the view calls `objectValueForTableColumn_byItem_()`. Again, it repeats the call only for each visible rows and columns. So, if a branch entry has two rows of children and there are two columns, the method `objectValueForTableColumn_byItem_()` gets called four times. Afterwards, the view repeats calling `isItemExpandable_()` for each visible child item and displays the disclosure icon when appropriate.

The data model

Listing 2 shows the main code for the script object `GaiyoModel`. Like most ASOC objects, `GaiyoModel` uses `NSObject` as its parent class. Plus, it declares the property `bufferDict` as an instance of `NSMutableDictionary`.

The `initialize()` handler prepares the test date tree to be shown on the outline view. It begins by calling the factory method `mainBundle()`, which returns an instance of `NSBundle`. Then it uses the instance method `pathForResource_ofType_()` to get the path to the property-list file `TreeDict.plist`. Next, the handler uses the factory method `dictionaryWithContentsOfFile_()` to load the data tree. It stores the data tree, now an instance of `NSMutableDictionary` into the property `bufferDict`.

Listing 2. Preparing the data model.

```

script GaiyoModel

— PROPERTIES:BASE
property parent : 'class "NSObject"'

— PROPERTIES:INSTANCE
property bufferDict : class "NSMutableDictionary"

on initialize()
  local tPth, tApp
  local tTmp

  — set the location of the first data file
  set tApp to mainBundle() of class "NSBundle" ~
    of the current application
  tell tApp
    set tPth to pathForResource_ofType_ ("TreeDict",
    "plist")
  end tell — tApp

  — load the data file
  — data:tree:dictionary
  tell class "NSMutableDictionary" of the current
  application
    set tTmp to dictionaryWithContentsOfFile_(tPth)
  end tell — class "NSMutableDictionary" of the current application
  set bufferDict to tTmp

end initialize
  
```

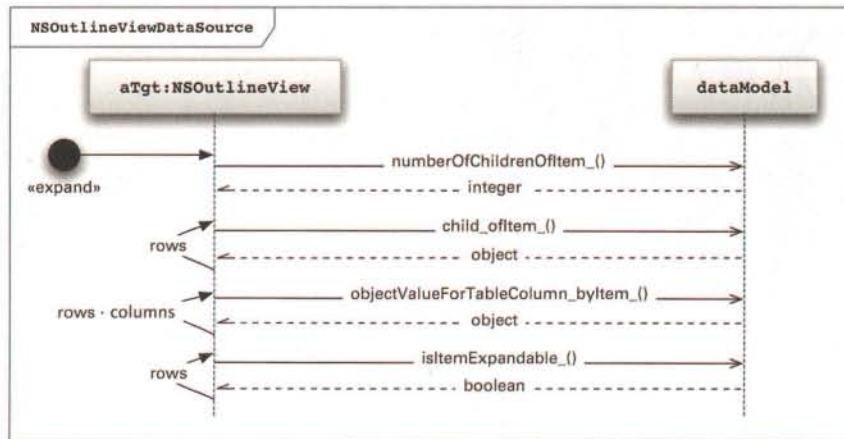



Figure 8. Displaying the child entries.

— continued in Listings 3 to 7

— ...

end script — GaiyoModel

The protocol handlers

Listing 3 shows how the GaiyoModel object implements the four data-source protocol handlers. Here again for brevity, we refer to each handler without the opening term `outlineView_`.

First up, the handler `numberOfChildrenOfItem_()` calls the tree routine `countChildren()` to find out if the given entry `aNod` has any valid children. Then the handler `child_ofItem_()` invokes the instance method `levelForItem_()`, passing along the entry `aNod`. That instance method returns the level of that entry as an integer. Next, the handler uses the tree routine `getChild()` to retrieve the child items for that entry at the row index `aIdx`. The routine returns the items as an `NSArray` instance, and the handler extracts the right child with the method `objectAtIndex_()`. If the retrieval fails, the handler assumes a null string as the child item.

Listing 3. Handling the protocol.

```

— Return the number of children for the given node
on outlineView_numberOfChildrenOfItem_(aTgt, aNod)
    local tCnt

    — retrieve the child count
    set tCnt to countChildren(aTgt, aNod)

    return (tCnt)
end outlineView_numberOfChildrenOfItem_

— Return the children for the given index and node
on outlineView_child_ofItem_(aTgt, aIdx, aNod)
    local tLev, tKey, tLst

    — identify the data level
    set tLev to levelForItem_(aNod) of aTgt

    — retrieve the child items
    try
        set tLst to getChild(aTgt, tLev, aNod)
        set tKey to objectAtIndex_(aIdx) of tLst
    on error
        set tKey to |string|() of class "NSString" of current
    application
    end try

```

```

        return (tKey)
    end outlineView_child_ofItem_

    — Does the node has any children?
    on outlineView_isItemExpandable_(aTgt, aNod)
        local tChk

        — retrieve the child count
        set tChk to countChildren(aTgt, aNod)
        set tChk to (tChk > 0)

        return (tChk)
    end outlineView_isItemExpandable_

    — Return the data item for the given outline column
    on outlineView_objectValueForTableColumn_byItem_(aTgt, aCol,
aNod)
        local tID, tVal
        local tChk

        — check the data level
        set tLev to levelForItem_(aNod) of aTgt

        if (tLev = -1) then
            — set the root item
            set tVal to "Animalia"
        else
            — identify the target outline column
            set tID to |identifier|() of aCol as string
            if (tID is "grup") then
                — redisplay the item
                set tVal to aNod
            else if (tID is "desc") then
                — display the item attribute
                set tChk to isExpandable_(aNod) of aTgt
                if (tChk) then
                    set tVal to ""
                else
                    set tVal to getDesc(aTgt, tLev, aNod)
                end if —(tChk)
            else
                set tVal to "n/a"
                end if —(tID is "grup")
            end if —(tLev = -1)

            return (tVal)
        end outlineView_objectValueForTableColumn_byItem_

```

The protocol handler `isItemExpandable_()` also calls `countChildren()` to find out if the entry `aNod` has at least one child. It returns its check result as a Boolean. The handler `objectValueForTableColumn_byItem_()` checks the level of the

KNOW...BEFORE IT'S TOO LATE.

THE BACKBONE OF YOUR MAC® BASED
MANAGED SERVICES PLATFORM

Client Group: Pretendco Computer Name: Robin Banks' Mac Pro



Hard Drive Health

No disk errors detected.
Within capacity: boot disk 65% full.



RAM Health

No trouble found.



RAID Status

Apple®RAID Card, Disk Utility RAID,
SoftRAID - No trouble found.



Time Machine Status

39 minutes since last successful backup.



Server Status

Billings Pro, Daylite, CrashPlan PRO,
LightSpeed POS - No trouble found.



Check Malware

No malware reported.



- Zero configuration
- Cost effective
- At-a-glance reporting
- Email alerts
- Custom branding available
(white label by default)

Pricing information at watchmanmonitoring.com/mactech

WATCHMAN
MONITORING



www.watchmanmonitoring.com

entry aNod. If the level is a -1, the handler returns the string "Animalia" to mark the root node. Otherwise, the handler uses the instance method identifier() to check which outline column to update. For the Group column (id 'grup'), it returns the value in aNod.

For the Description column (id 'desc'), the handler checks if aNod is a child item using the instance method isExpandable_(). If the check proves true, the handler uses the tree routine getDesc() to retrieve the specified attribute. If not, it returns a null string.

The tree routines

The GaiyoModel object uses four routines to retrieve specific items from its bufferDict property. All four gets the outline view as one of its arguments.

The routine countChildren() (Listing 4) begins by checking the level of the branch item aNod. If the level is a -1, the handler returns the number of root entries held by bufferDict. Otherwise, the routine uses getChild() to drill down to the specified branch. Then it returns the number of child items that branch may contain.

Listing 4. Counting the child items.

```
— Check if a given node has children
to countChildren(aTgt, aNod)
    local tCnt, tNod, tLev

    — identify the data level
    set tLev to levelForItem_(aNod) of aTgt

    if (tLev = -1) then
        set tCnt to |count|() of bufferDict
    else
        — get the number of children
        set tNod to getChild(aTgt, tLev, aNod)
        if (tNod is missing value) then
            set tCnt to 0
        else
            set tCnt to |count|() of tNod
        end if — (tNod is missing value)
    end if — (tLev=-1)

    — return the count result
    return (tCnt)
end countChildren
```

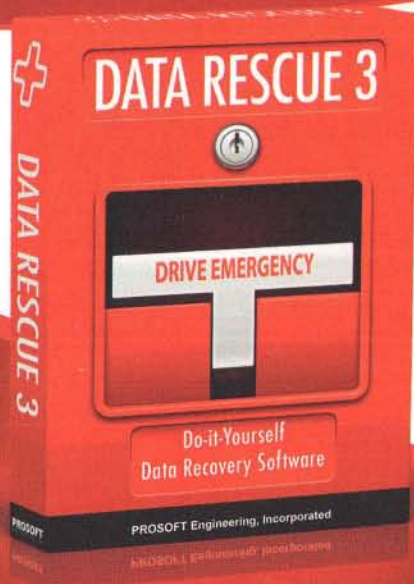
The routine getChild() (Listing 5) starts with a call to its fellow routine path2item(). It gets an AppleScript list object in return. Then getChild() counts the number of branches it needs to traverse. It uses a repeat loop to walk down the data tree, and uses the instance method valueForKey_() to move to the next branch. Once reading the desired branch, the routine uses the instance method allKeys() to read its child items.

Listing 5. Retrieving the child item.

```
— Return the chosen node
to getChild(aTgt, aLev, aNod)
    local tLst, tCnt, tIdx
    local tNod, tBuf
    local tChk

    — retrieve the node path
    set tLst to path2Item(aTgt, aLev, aNod)

    — retrieve the node item
    set tCnt to count of tLst
    set tBuf to bufferDict
```

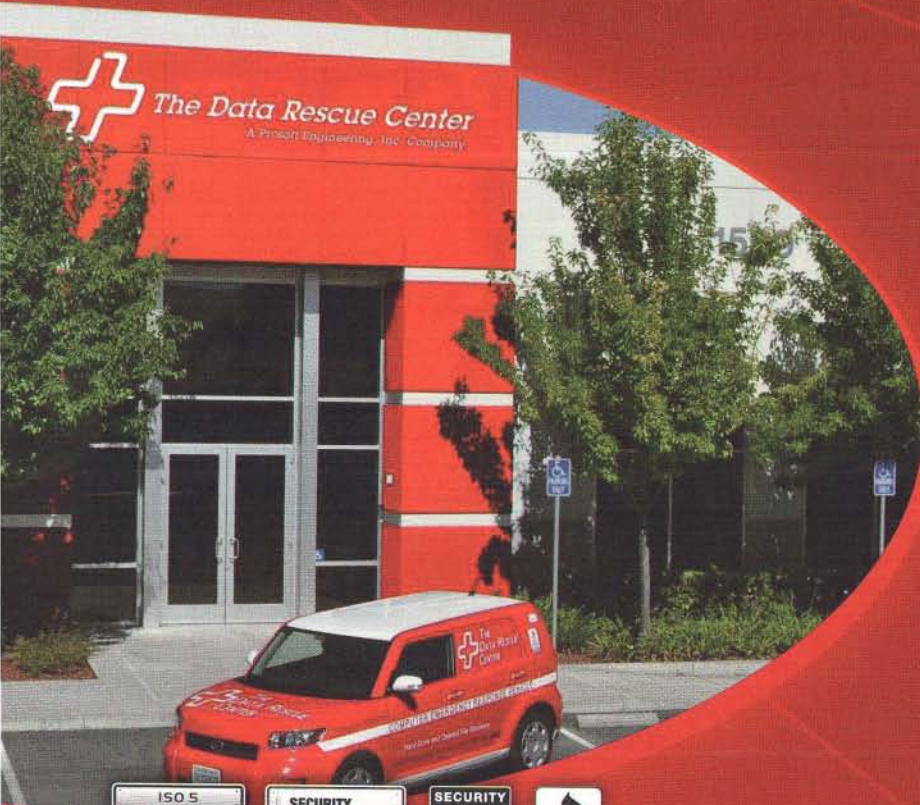



From the makers of...

Data Rescue 3*

The best selling data recovery software for Mac...

...comes the
complete data recovery service.



The Data Rescue Center was founded by Prosoft Engineering, makers of the award-winning Data Rescue software. Our recovery know-how provides unparalleled advantages over other recovery services.

- FREE hardware diagnosis
- FREE specialized drive & laptop boxes sent direct to you at NO COST
- No Data, No Charge
- Professional Results at a Lower Cost
- Numerous awards in 2010 for Best Computer Recovery Service
- Class 100/ISO 5 Cleanroom
- Class 2 Vault
- FREE E-Waste Recycling

The Data Rescue Center is headquartered in Livermore, California at a new state-of-the-art facility near Lawrence Livermore National Laboratory.

This new facility provides our recovery engineers with the top technology to recover your computer data while offering best-in-class security features to protect your data from being compromised.



The Data Rescue Center

hard drive recovery · data migration · photo archiving

1599 Greenville Rd | Livermore CA 94550

877-501-4949

TheDataRescueCenter.com



Register

**Get your .COM
or any other
domain name
here!**

FREE with every domain:

- **FREE!** Starter Web Page
- **FREE!** Getting Started Guide
- **FREE!** Complete Email
- **FREE!** Change of Registration
- **FREE!** Parked Page w/ Domain
- **FREE!** Domain Name Locking
- **FREE!** Status Alert
- **FREE!** Total DNS Control

Just visit

www.mactechdomains.com
to register for your domain today!

**Starting
at
\$1.99**

**when a non-domain name product
is purchased. Limitations apply.**

```
repeat with tIdx from 1 to tCnt
    set tNod to item tIdx of tLst
    set tBuf to valueForKey_(tNod) of tBuf
end repeat — with tIdx from 1 to tCnt
```

```
set tLst to allKeys() of tBuf
```

```
— return the retrieval results
return (tLst)
end getChild
```

The routine `getDesc()` (Listing 6) also starts with a call to `path2Item()`. It too uses the instance method `valueForKey_()` to drill down the data tree held by `bufferDict`. But this time, the routine converts the value held by the leaf item `tBuf` into a string. And if `tBuf` holds an empty list, the routine returns the string “—” as its result.

Listing 6. Retrieving the description.

```
— Return a description for the given item
to getDesc(aTgt, aLev, aItm)
    local tDes, tPth

    — retrieve the item path
    set tPth to path2Item(aTgt, aLev, aItm)

    — retrieve the item description
    set tCnt to count of tPth
    set tBuf to valueForKey_(tNod) of tBuf
    end repeat — with tIdx from 1 to tCnt

    if (tBuf = {}) then
        set tDes to “—”
    else
        set tDes to tBuf as string
    end if — (tBuf is missing value)

    — return the description result
    return (tDes)
end getDesc
```

Finally, the routine `path2Item()` (Listing 7) traces the path from the root node to the given tree item. First, it starts with the tree item set inside an AppleScript list. Then using the instance method `parentForItem_()` from `NSOutlineView`, the routine works its way back to the root node. As it does, the routine adds each parent branch to the list. Then, upon reaching the root node, the routine returns the complete list to the calling routine.

Listing 7. Walking the branches.

```
— Return the path to a given item
to path2Item(aTgt, aLev, aItm)
    local tPth, tPrv, tStr, tCnt

    — initialize the item path
    set tPth to {}
    set tPth to (aItm as list) & tPth

    set tPrv to aItm
    repeat until (tPrv is missing value)
        set tPrv to parentForItem_(tPrv) of aTgt
        set tPth to (tPrv as list) & tPth
    end repeat — until (tPrv is missing value)

    — finalize the path
    set tCnt to count of tPth
    if (tCnt is 1) then
        set tPth to {}
    end if
```



```

else
    set tPth to items 2 thru tCnt of tPth
end if — (tCnt is 1)

— return the retrieved path
return (tPth)
end path2Item

```

The sample run

Figure 9 shows how Gaiyo displays the four branches *adjacent* to the root node. Since branch *rotifera* has no valid children, it is the only entry without a disclosure icon. But note that the three disclosure icons are set to their collapsed state.



Figure 9. The root branches.

Next, Figure 10 shows branch *mollusca* expanded to show its two children *bivalvia* and *cephalopoda*. As those two children have children of their own, they appear with the required disclosure icon.

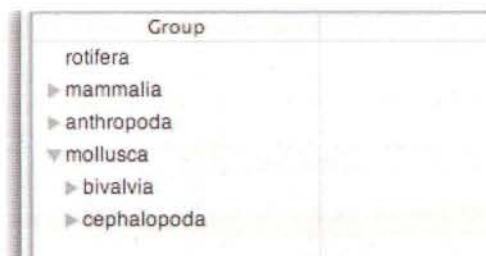


Figure 10. Child items for branch mollusca.

Figure 11 shows the branch *bivalvia* expanded to show its four children—all leaf items. Note how only the leaf items get entries under the *Description* column. Branch items, on the other hand, get none.

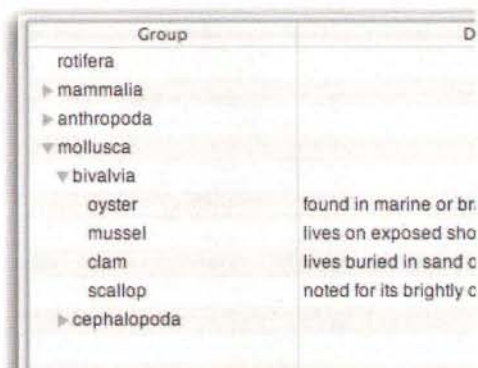


Figure 11. Child items for branch bivalvia.



CYBORG R.A.T. 7
GAMING MOUSE FOR PC & MAC
CONTAGION


6400 dpi

available at apple.com

DPI range	25-6400DPI (25DPI steps)
Acceleration	Up to 50G
Polling Rate	Dynamic up to 1000Hz
Tracking Speed	Up to 6m/sec
Always On	Yes
"Slick" PTFE Feet	Yes
Gold-plated connector	Yes
Braided cable	Yes
L.E.D. colors	for each mode

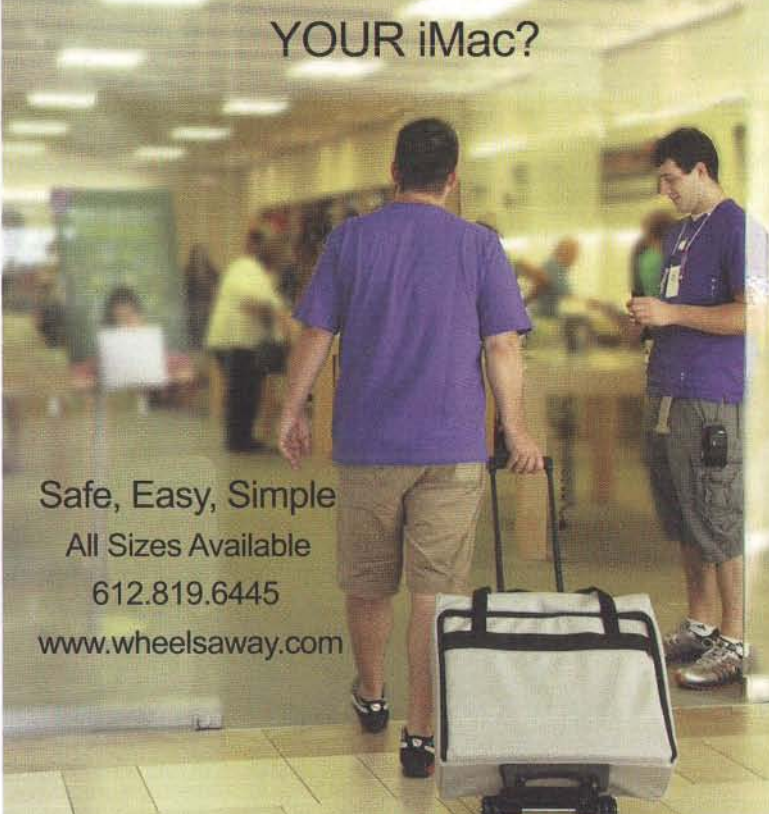
Best Gaming Mouse Ever
PC GAMER 98 EDITORS' CHOICE
NEW for Mac

© 2011 Mad Catz Interactive Asia Limited (MCIA)
cyborggaming.com



Wheelsaway

How Do You Transport
YOUR iMac?



Safe, Easy, Simple
All Sizes Available
612.819.6445
www.wheelsaway.com

Wrapping Up

The `NSOutlineView` class lets us present data that have a tree-like structure. It can differentiate between data items that form branches and those that form leaves. It can also show which items are grouped under which branch.

This article has shown us how to use `NSOutlineView` with ASOC. We learned about its data source protocol and how to implement said protocol as AppleScript handlers. We learned how to load the data tree from a property-list file. And we learned how to walk the tree and retrieve specific items from its store for display.

So ends our study of the `NSOutlineView`. But remember, there is a different way to present data trees, which is with a *browser view*. Use the control that best suits your needs.

Till next month, I bid you well.

Bibliography and References

- Apple Developer Connection. *NSOutlineView Class Reference*. MacOS X Developer Library. Apple, Inc. 2010 May. [Online]. Available:
http://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/Classes/NSOutlineView_Class/NSOutlineView_Class.pdf.
- Apple Developer Connection. *NSOutlineViewDelegate Protocol Reference*. MacOS X Developer Library. Apple, Inc. 2010

Jan. [Online]. Available:

http://developer.apple.com/library/mac/documentation/Cocoa/Reference/NSOutlineViewDelegate_Protocol/NSOutlineViewDelegate_Protocol.pdf.

Apple Developer Connection. *Outline View Programming Topics*. MacOS X Developer Library. Apple, Inc. 2010 Mar. [Online]. Available:

<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/OutlineView/OutlineView.pdf>.

CocoaDev. *NSOutlineView*. Internet:

<http://www.cocoa-dev.com/index.pl?NSOutlineView>. 2009 Jan [2011 Mar].

MI



About The Author

JC is a freelance engineering writer from North Vancouver, British Columbia. He frequently contributes articles to MacTech and REALbasic Developer. He also wrote for the now defunct Python Magazine, and is now working on a database e-Book. When away from the writing pile,

JC spends quality time with his foster nephew, as a proper uncle should. He can be reached at anarakisware-at-gmail-dot-com.

Take Control of your Software

Manage all your Mac, Windows and iOS devices from a single interface.

filewave



- File level deployment
- Patch Management
- Schedule activation/deactivation of client systems
- Easily revert to previous versions of software
- Post deployment control of any file on your client systems

explore more at filewave.com



BookEndz[®]

Docking Stations for Apple Computers

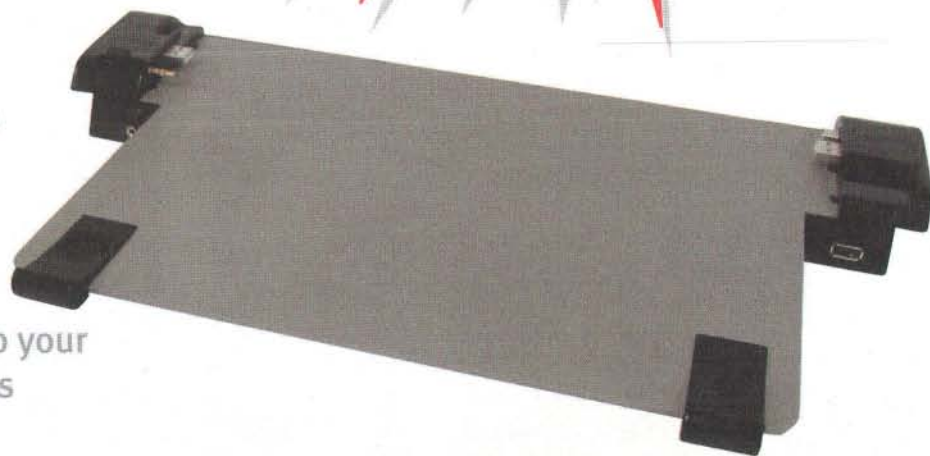


DOCKING STATIONS FOR APPLE COMPUTERS

Turns your laptop into a desktop



NEW!
MacBook Air
11" and 13"
Docking stations



- Includes USB hub to give you a total of 4 USB 2.0 ports
- Allows access to SD slot
- Adds an Ethernet Connection to your MacBook Air for wired networks
- Allows you to use an external monitor in addition to your laptop
- Angles your MacBook Air off a flat surface

Visit our website for latest product announcement www.BookEndzdocks.com



BookEndz[®]

Manufactured by Olympic Controls

1250 Crispin Drive • Elgin, Illinois 60123 • USA

Phone: 847-742-3566 • Fax: 847-742-5686 • Toll Free: 888-622-1199 • E-mail: Sales@BookEndzdocks.com

Apple Software Updates with Reposado

An open-source alternative to Apple's Software Update service

by Greg Neagle, *MacEnterprise.org*



MacEnterprise.org

Mac OS X enterprise deployment project

Introduction

Many, if not most Mac OS X administrators are looking to reduce their dependency on Mac OS X Server software and hardware. Many services provided via OS X Server are easily replicated on other hardware and/or software: DHCP, DNS, LDAP, web serving, file serving, and email, as examples, are very straightforward to implement on alternate server platforms like Linux or Windows.

However, there are a few Apple-specific services that have been a bit trickier to implement outside of OS X Server. This month, we look at a replacement for Apple's Software Update service.

Reposado is a set of tools written in Python that replicate the key functionality of Mac OS X Server's Software Update service. Created and released by Walt Disney Animation Studios, Reposado is open source, licensed under the new BSD license. Reposado allows you to run a local Apple Software Update server on the hardware and OS of your choice.

Reposado Capabilities

Reposado, like the Apple Software Update service it replaces, can cache Apple updates on a local web server, saving on Internet bandwidth usage for your organization.

Also, like Apple's service, Reposado allows an administrator to approve updates before they are offered to managed clients.

Additionally, Reposado enables an administrator to create any arbitrary number of "branches" of the Apple catalogs. These branches can contain any subset of the available updates. For example, one could create "testing" and "release" branches, and then set some clients to use the "testing" branch catalog to test newly released updates. Other clients (the majority of clients, in fact) would use the "release" branch catalog, which would contain updates that had been through the testing process.

If you configure Reposado to also download the actual updates as well as the catalogs, you can continue to offer

updates that have been superseded by more recent updates. For example, if you are currently offering the Mac OS X 10.7.2 update to your clients, and Apple releases a 10.7.3 update, you can continue to offer the (now deprecated) 10.7.2 update until you are ready to release the newer update to your clients. You can even offer the 10.7.2 update to your "release" clients while offering the 10.7.3 update to your "testing" clients. Offering "deprecated" Apple Software Updates is a feature that is difficult with Apple's tools.

Getting Started with Reposado

What you need

The Reposado tools.

Python 2.5–2.7 with plistlib. (Reposado has been tested with Python 2.6, but should work with 2.5–2.7 as long as plistlib is available. plistlib was included as a Mac-specific library with Python 2.5, and for all platforms with Python 2.6.)

The `curl` binary (this is a standard tool on Mac OS X, and available for most other OSes).

A web server. Apache2 works well, but almost any modern web server should suffice.

Storage space for the catalogs and update packages. If you are replicating all update packages for Tiger through Lion, you'll need approximately 90GB of space as of November 2011. The need for space will grow as Apple releases additional updates. If you are only replicating catalogs, you'll probably need less than 100MB of space, though the exact amount of space needed depends on the number of branch catalogs you create.

Getting the Reposado tools

If you are familiar with git, the easiest way to get a copy of the Reposado tools is to do a git clone from Reposado's GitHub site:


```
git clone https://github.com/wdas/reposado.git
```

Alternately, you may download the source from GitHub at this URL: <https://github.com/wdas/reposado/zipball/master>. The tools are in the `code` directory. The tools (which are just Python scripts) do not need to be any place in particular to operate; you may put them wherever you like.

Setting up Reposado

1. Create a directory in which to store replicated catalogs and updates, and another to store Reposado metadata. These may share a parent directory, like so:

```
/Volumes/data/reposado/html  
/Volumes/data/reposado/metadata
```

Make sure you have enough space for the replicated catalogs and updates. Make sure these directories are writable by the user `repo_sync` will run as, and readable by the user your webserver runs as.

2. Configure your web server to serve the contents of the updates root directory you created (`/Volumes/data/reposado/html` in the example above). If you are planning to replicate and serve the actual update packages as well as the catalogs, make note of the URL needed to access the updates root directory via HTTP. This will be the "Base URL for your local Software Update Service" when configuring Reposado in the next step.

3. `cd` to the directory containing the Reposado tools. Configure Reposado by running:

```
./repoutil -configure
```

You'll be asked three questions:

Path to store replicated catalogs and updates:

This corresponds to the path `/Volumes/data/reposado/html` in step 1 above.

Path to store Reposado metadata:

This corresponds to the path `/Volumes/data/reposado/metadata` in step 1 above.

Base URL for your local Software Update Service
(Example: `http://su.your.org` - leave empty if you are not replicating updates):

This is the URL you configured in step 2 above.

4. Run `./repo_sync` to replicate Apple catalogs and update packages to your Reposado server. The first time you do this it may take several hours to complete if you are replicating packages as well as catalogs. (Or even more time if you have a slow connection to the Internet.)

5. Test your work so far by visiting a catalog URL in your browser. If `http://su.myorg.com` is the URL for the updates root directory you created earlier, then

IMPORTANT THOUGHTS
SHOULD TAKE TIME TO FORM.
NOT TYPE.



GET UP TO SPEED WITH THE
ZAGG[®]folio[™]
FOR iPad[®] 2 AND GALAXY TAB[™] 10.1



- Thin, durable, carbon fiber textured folio with removable Bluetooth[™] wireless keyboard.

- Island style keys provide a comfortable, fast and accurate typing experience.

- Built-in stand system is ideal for on the go use. Supports tablet in both landscape and portrait mode.

- Special function keys operate convenient shortcut features directly from the keyboard.

ZAGG www.ZAGG.com NASDAQ:ZAGG Scan to learn more ▶



<http://su.myorg.com/content/catalogs/others/index-leopard-snowleopard.merged-1.sucatalog> is the Catalog URL for the Snow Leopard updates catalog. You should see a plist version of the updates catalog displayed in your browser.

6. Next test: run **softwareupdate** on a client, again pointing it to your Catalog URL:

```
softwareupdate -l -CatalogURL \
"http://su.myorg.com/content/catalogs/others/index-leopard-
snowleopard.merged-1.sucatalog"
```

If there are no errors, you've successfully configured Reposado and successfully replicated Apple Software Updates.

Configuring clients to use your Reposado server

If you've never used the Software Update Service on Mac OS X Server, you may be unfamiliar with configuring Mac OS X clients to use a Software Update Server other than Apple's server.

To configure a client machine to use your Reposado server, you must set the value of CatalogURL in **/Library/Preferences/com.apple.SoftwareUpdate.plist**. This is commonly done using the command-line **defaults** tool:

```
sudo defaults write \
/Library/Preferences/com.apple.SoftwareUpdate CatalogURL \
<catalog_url>
```

where **<catalog_url>** is the URL to the catalog file. The following lists example Catalog URLs for each major release of Mac OS X supported by Reposado:

Tiger Clients

<http://su.yourorg.com/content/catalogs/index.sucatalog>

Leopard Clients

<http://su.yourorg.com/content/catalogs/others/index-leopard.merged-1.sucatalog>

Snow Leopard Clients

<http://su.yourorg.com/content/catalogs/others/index-leopard-snowleopard.merged-1.sucatalog>

Lion Clients

<http://su.yourorg.com/content/catalogs/others/index-lion-snowleopard-leopard.merged-1.sucatalog>

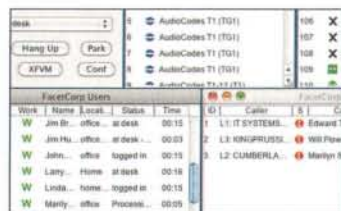
Besides using the **defaults** tool via a script, a payload-free package, or the "Send UNIX Command" facility in Apple Remote Desktop, you could also set the CatalogURL using Managed Preferences or MCX.

Implementing Branch Catalogs

In the examples above, the Catalog URLs point to the "raw" catalog as downloaded from Apple. This "raw" catalog features all current updates from Apple. If you point your client machines at these raw catalogs, your clients may not need to go over the Internet to get updates, but your clients will still get Apple's updates as soon as they are released by Apple, and without an opportunity for you to vet them. Apple's Software Update Service has the ability to allow administrators to approve only a subset of updates to be offered to clients. Reposado has that feature as well. Further, Reposado supports multiple "branch" catalogs that offer different sets of available updates. You can use this feature to implement a "unstable, testing, release" workflow where a very small number of clients ("unstable") get Apple updates as soon as they are released by



Your communication server
for up to 500 users.



Watch phone system activity and easily
control your calls with the FacetPhone
user interface on your Mac®.



Your iPhone®
or iPad® can
become an
extension
on the
FacetPhone
system.

The full featured
business phone
system that runs
on a Mac.

FACETPHONE

 **FacetCorp**

1.877.FacetGo
info@facetcorp.com
www.facetcorp.com

Apple; a larger group (but still a subset) of clients get the new updates next for testing, and finally, the updates are released to all clients.

To implement this workflow, you use the **repoutil** tool to create multiple branch catalogs. You then selectively add updates to one or more branch catalog. For example, let's say you create "testing" and "release" branches. When Apple releases a new update, it is available in the "raw" catalogs described earlier. You could configure a few of your client machines to use the "raw" catalogs. If nothing breaks after a day or so, you could then add the new updates to the "testing" branch catalog. You would then have a larger group of machines that were configured to get their Apple updates from the testing branch. Again, after a few days or a week, you could add the new Apple updates to the "release" branch. The majority of your client machines would be configured to use this "release" branch. This workflow enables you to gradually rollout new Apple updates so you have time to test and verify that the updates do not cause issues in your environment.

Creating and Managing Branches

To create branch catalogs you use **repoutil -new-branch**:

```
./repoutil -new-branch testing
./repoutil -new-branch release
```

These newly created branch catalogs are empty – you need to add updates (or in Apple parlance, "products") to the branch catalogs. Products are added using their product IDs. You can get a list of available product IDs using **repoutil -products**, which will print a list like this:

```
041-2790 MacBook Pro Video Update 1.0 2011-10-24
041-2859 iPhoto Update 9.2.1 2011-10-26
041-2472 iMac EFI Firmware Update 1.7 2011-10-26
041-2592 MacBook Air EFI Firmware Update 2.2 2011-10-26
041-2517 Mac mini EFI Firmware Update 1.4 2011-10-26
041-2515 MacBook Pro EFI Firmware Update 2.3 2011-10-26
041-2800 Thunderbolt Software Update 1.0 2011-10-27
(Deprecated)
041-2167 HP Printer Software Update 2.8 2011-10-27
041-2305 EPSON Printer Software Update 2.9 2011-10-27
041-2856 Aperture Update 3.2.1 2011-10-27
041-3123 Thunderbolt Software Update 1.1 2011-10-28
041-3149 Thunderbolt Firmware Update 1.0 2011-11-07
041-1940 Java for Mac OS X 10.7 Update 1 1.0 2011-11-08
041-1943 Java for Mac OS X 10.6 Update 6 6.0 2011-11-08
```

You could then add the two new Java updates to the testing catalog like so:

```
./repoutil -add-product 041-1940 041-1943 testing
Adding 041-1940 (Java for Mac OS X 10.7 Update 1-1.0) to
branch testing...
Adding 041-1943 (Java for Mac OS X 10.6 Update 6-6.0) to
branch testing...
```

After you are happy with the testing period, you'd add them to the release catalog as well:

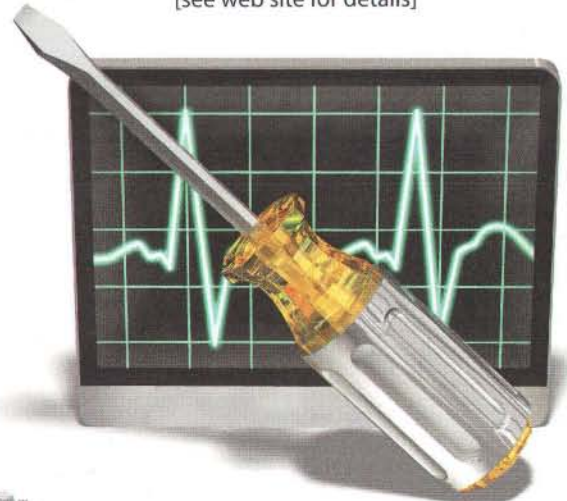
MACTECH

TECHTOOL PRO 6

Total Diagnostics, Maintenance and Drive Repair

AppleCare users - you qualify to upgrade your
TechTool Deluxe to TechTool Pro 6
today for only \$39.99!

[see web site for details]



MICROMAT
SINCE 1989

707.566.3831
www.micromat.com

FMC
ONLINE • ONSITE • INCLASS

 **Authorized
Training Center
Gold Level**

**Lion and iOS 5 Classes
NOW FORMING**

Certified Trainers and Curriculum

Small Class Sizes

Corporate Training Programs

Online Instructor-Led Training

Certifications & Exams



877.362.8724

FMCTraining.com

New York | Boston | Philadelphia
Washington DC | Orlando | Chicago | Dubai
Onsite Training Worldwide


```
./repoutil -add-product 041-1940 041-1943 release
Adding 041-1940 (Java for Mac OS X 10.7 Update 1-1.0) to
branch release...
Adding 041-1943 (Java for Mac OS X 10.6 Update 6-6.0) to
branch release...
```

Deprecated Updates

In the list of products above, you might notice one – Thunderbolt Software Update 1.0 – has been marked as “Deprecated”. This means it is no longer offered in any Apple catalog. As long as Reposado has replicated the item to its local storage, you can continue to offer it in your catalogs. It’s often useful to continue offering a deprecated product (like the OS X 10.7.1 update) in the “release” branch at the same time you are testing its replacement (for example, the OS X 10.7.2 update) in the “testing” branch. But eventually you will probably want to remove the deprecated update from a branch:

```
./repoutil -remove-product 041-2800 testing
Removing 041-2800 (Thunderbolt Software Update-1.0) from
branch testing...
```

Often deprecated products have been replaced by updated versions. That’s the case here, so we might want to add the replacement product to our testing branch:

```
./repoutil -add-product 041-3123 testing
Adding 041-3123 (Thunderbolt Software Update-1.1) to branch
testing...
```

Adding and removing items from branch catalogs can become tedious. If you are using a testing/release workflow and would like to promote everything in testing to release, there’s a handy shortcut: `./repoutil -copy-branch`, which copies all the contents of one branch to another:

```
./repoutil -copy-branch testing release
```

This is a fast way to move everything from testing to release.

Configuring Clients to use Branch Catalogs

Once you’ve created branch catalogs, you’ll want to configure your clients to use one or another branch. The Catalog URLs follow a simple rule: for each “raw” Catalog URL, branch catalogs append an underscore and the branch name at the end of the catalog name, but before the “.sucatalog” extension. If the “raw” Tiger catalog is:

<http://su.yourorg.com/content/catalogs/index.sucatalog>

then the testing branch for Tiger clients is:

http://su.yourorg.com/content/catalogs/index_testing.sucatalog

and the release branch is:

http://su.yourorg.com/content/catalogs/index_release.sucatalog

Introducing a better hosting solution.

- ✓ Running CloudLinux
- ✓ Limited Accounts Per Server
- ✓ Full cPanel With Each Account
- ✓ Daily Backups
- ✓ Free Domain and Wildcard SSL

Discounts on our UBB Forum



- ▶ In use since 1997
- ▶ Easy to setup
- ▶ Easy to administer
- ▶ Built in PhotoGallery

As seen on the MacTech Forums
at applecentral.com

Visit www.mindraven.com and let us show you how we can give your website a better home with our shared hosting plans, semi-dedicated and dedicated servers.



MINDRAVEN.com

Additional repoutil commands

repoutil has a few more tricks:

repoutil -branches lists all currently available branch catalogs.

repoutil -delete-branch <branch_name> removes a branch.

repoutil -list-branch <branch_name> lists all the products in a branch.

repoutil -deprecated lists all deprecated updates.

repoutil -product-info <product_id> prints available detail for <product_id>.

Caveats and Conclusions

Reposado does not offer some features of Apple's Software Update service.

Apple's Software Update service includes an embedded web server; Reposado requires you configure a separate web server. But if you are running Reposado on a machine that already has a web server process running (like Apache2) you can simply use that.

Unlike the Software Update module in Apple's Server Admin application, Reposado has no GUI. All Reposado setup and management is done via the command line. The lack of a GUI, however, actually makes it easier to get Reposado up and running on a wide range of OSes and hardware.

Finally, Apple's software update offering automatically checks every day for new updates. Since Reposado is not tied to a particular OS, it's up to you to set things up so that `repo_sync` runs periodically - each OS handles this sort of thing differently.

These caveats aside, Reposado should be of interest to any Mac OS X administrator who wants to lessen his or her organization's dependency on OS X Server, who would like to implement a testing/release workflow for Apple updates, or who needs to continue to offer "deprecated" updates indefinitely. Visit Reposado's GitHub site at <https://github.com/wdas/reposado>, and learn more about Walt Disney Animation Studio's open source efforts at <http://www.disneyanimation.com/technology/opensource>.

MI

About The Author

Greg Neagle is a member of MacEnterprise (macenterprise.org) and is a senior systems engineer at a large animation studio. Greg has been working with the Mac since 1984, and with Mac OS X since its release. Greg Neagle and Edward Marzcek's book: *Enterprise Mac Managed Preferences*, which covers Apple's Managed Preferences, was recently published by Apress. Greg can be reached at gregneagle@mac.com.

MACTECH

STOP SHARING!



START FAXING!

**Each subscriber receives
faxes directly by email
as PDF file attachments.**

**Corporate accounts from
3 to 100+ users available**

**For more information
and a special offer for
MacTech readers, visit**

www.MaxEmail.com/MacTech

maxemail®

Call: 800-964-2793

Buyers' Guide

Listing by Company

Acquia, Inc.

150 Presidential Way, Suite 310
Woburn, MA 01801
Sales: 888-9-ACQUIA
Main web site: www.acquia.com

Acquia gives you answers, tools, and support to make Drupal even better.

See advertisement on page 35 of this issue.

Alice Dev Team

Main web site: www.alicedev.com

We are a small, professional team of Apple Developers from the heart of snow-covered Siberia. We design and develop utilities, useful tools, entertainment and social apps for Mac, iPhone and iPad. Our aim is to create the apps that give you joy and put a smile on your face!

See advertisement on page 14 of this issue.

Audioengine

555 Nathan Rd
Kowloon,
Hong Kong
Main web site: www.audioengineusa.com

Audioengine designs and builds innovative audio products with all your music in mind. Great sound, simple but elegant designs, high-quality materials, and truly useful features are what the Audioengine Advantage is all about. For information about our products go to: www.audioengineusa.com Award-winning Sound and the Perfect Audio Upgrade.

See advertisement on page 28 of this issue.

Background Backup, Inc.

209-99 Fifth Avenue
Ottawa, ON K1S 5P5
Sales: (866) 838-IRON (4766)
Main web site: www.irongate.ca

Server management and consulting

See advertisement on page 18 of this issue.

Benchmark Email

1777 Bellflower Blvd., Suite 100
Long Beach, CA 90815
Sales: 800.430.4095
Main web site: www.benchmarkemail.com

Build relationships. Make sales. Promote your goods. Gather data. Benchmark's email marketing software delivers powerful, user-friendly and affordable tools to create, send and track high-performance email and newsletter campaigns.

See advertisement on page 45 of this issue.

BITS Limited

514 Cathedral Drive
Aptos, CA 95003
Main web site: www.bitsltd.net

As many individuals and companies are trying to reduce their carbon footprints we, as always, are trying to assist them because we know how important energy conservation is to the planet and every individual.

See advertisement on page 10 of this issue.

Black Magic Design

71 Thistlethwaite Street
South Melbourne VIC 3205,
Australia
Sales: 408 954 0500
Main web site: www.blackmagic-design.com

Blackmagic Design creates the world's highest quality video editing products, color correctors, video converters, video monitoring, routers, live production switchers, disk recorders, waveform monitors and film restoration software for the feature film, post-production and television broadcast industries.

See advertisement on page 17 of this issue.

Brad Sniderman

23679 Calabasas Road #558
Calabasas, CA 91302
Sales: 818-706-0631
Main web site: www.sniderman.com

I was admitted to the State Bar of California in 1995 and am now working as a sole practitioner, focusing in Wills and Trusts, Intellectual Property, Business, Commercial and Corporate Law.

See advertisement on page 14 of this issue.

CableJive

380 Pleasant St, Ste 25
Malden, MA 02148
Sales: (781) 350-3055
Main web site: cablejive.com

See advertisement on page 24 of this issue.

codefortytwo software

1 Main St. SE #400
Minneapolis, MN 55414-1035
Main web site: www.code42.com

Our latest invention is CrashPlan, the most simple, most reliable and smartest backup on earth. A pretty hefty claim, but true! CrashPlan compresses, encrypts and automatically transmits your files offsite. What do you say? Others do that too? Not like this they don't! This bad boy has been cooking in our labs for years!

See advertisement on page 19 of this issue.

ComputerTree

1760 Jonestown Rd.
Winston-Salem, NC 27103
Sales: (800) 467-9820
Main web site: www.computertree.com

Since 1982, ComputerTree has been a trusted partner, consultant and resource to the publishing, retail, entertainment, new media and education industries. We offer turnkey solutions and experience in cross-platform integration, corporate communications, marketing, content management, device management systems and deployment services

See advertisement on page 18 of this issue.

Dartware LLC

66 Benning Street, Suite 7
West Lebanon, NH 03784
Sales: 877.276.6903
Main web site: www.dartware.com

InterMapper(r) is a leading network monitoring, mapping and alerting application. It's an easy to configure and fully featured management tool with an integrated robust NetFlow analyzer. These innovative tools earn quick ROI by proactively notifying administrators to potential hardware, software and bandwidth issues that could cause business interruptions.

See advertisement on page 13 of this issue.

Data Jack

14911 Quorum Dr., Ste 140
Dallas, TX 75254
Sales: 1-888-693-4522
Main web site: www.datajack.com

DataJack offers a range of flexible and affordable data plans with no contract or activation fees.

See advertisement on page 41 of this issue.

DriveSavers Data Recovery

400 Bel Marin Keys Blvd
Novato, CA 94949
Sales: 800-440-1904
Main web site: www.drivesavers.com

The ultimate data recovery service.

See advertisement on page 34 of this issue.

e3 Software

465 Fairchild Drive, Suite 229
Mountain View, CA 94043
Main web site: ethreesoftware.com

Powerful email marketing software for your Mac.

See advertisement on page 21 of this issue.

MACTECH

Facet Corp.

5999 Summerside Dr.
Ste 210
Dallas, TX 75252
Sales: 972.985.9901
Main web site: www.facetcorp.com

FacetPhone reduces costs, increases productivity, and seamlessly connects remote offices and teleworkers. Learn how your business or contact center can benefit from a feature rich, highly scalable FacetPhone system. Now with the FacetPhone server software running on Mac OS X, this is THE phone system for Apple enthusiasts!

See advertisement on page 58 of this issue.

Faronics

620-609 Granville St.
Vancouver, BC V7Y 1G5
Main web site: www.fmctraining.com

Deep Freeze instantly protects and preserves baseline computer configurations. No matter what changes a user makes to a workstation, simply restart to eradicate all changes and reset the computer to its original state - right down to the last byte. Expensive computer assets are kept running at 100% capacity and technical support time is reduced or eliminated completely.

See advertisement on page 33 of this issue.

Fiberlink

1787 Sentry Parkway West
Building 18, Ste 200
Blue Bell, PA 19422
Sales: 1.855.maas360
Main web site: www.maas360.com

With 20 years of experience in delivering enterprise mobility management solutions, Fiberlink has the in-depth functional and technical expertise necessary to accelerate deployment, reduce risk, increase employee productivity, and simplify mobile device management.

See advertisement on page 7 of this issue.

FileWave (USA), Inc.

10714 Square Praurue Dr.
Fishers, IN
46038-7815
Main web site: www.filewave.com

See advertisement on page 54 of this issue.

Future Media Concepts

299 Broadway,
Suite 1510
New York, NY 10007
Sales: 877-362-8724
Main web site: www.fmctraining.com

Future Media Concepts, Inc., the nation's premier digital media training center, provides manufacturer-authorized training in all areas of digital media including digital video and film editing, web design and development, sound design, DVD authoring, 3D animation, motion graphics, desktop publishing and Mac IT. Certification testing and on-site training are available.

See advertisement on page 59 of this issue.

Gefen Inc.

20600 Nordhoff Street
Chatsworth, CA 91311
Sales: (800) 545-6900
Main web site: www.gefen.com

Gefen supplies a wide selection of signal switchers, splitters, extenders, scalars, converters, KVM solutions and home theater accessories that support digital signage, education, government, residential, retail, industrial and commercial applications. Gefen's hardware enables audio/video and computer systems to be easily integrated, extended, distributed and optimized to maximize performance.

See advertisement on page 12 of this issue.

Hansaworld

1501 Broadway Ste 12068
New York, NY 10036
Sales: 44 (0) 845 123 2732
Main web site: www.hansaworld.com

HansaWorld provides a single, integrated ERP software solution covering Accounts, Enterprise Resource Planning and Customer Relationship Management. Unusual for an ERP vendor, HansaWorld has also developed fully integrated vertical market ERP software solutions for hotels, restaurants, retail, professional services, manufacturing, membership, rental, training companies and others.

See advertisement on back cover of this issue.

HostGator

11251 Northwest Freeway, Suite 400
Houston, TX 77092
Sales: 866.964.2867
Main web site: www.hostgator.com

Web hosting made easy and affordable

See advertisement on page 39 of this issue.

iDeveloper TV

4164 Austin Bluffs Parkway, Suite 157
Colorado Springs, CO 80918
Main web site: www.zarrastudios.com

Zarra Studios was started by Marcus S. Zarra in the summer of 2005. While Marcus is based out of Colorado Springs, Colorado, Zarra Studios is distributed with developers in several locations

See advertisement on page 38 of this issue.

IGC, Inc. / MaxEMail.com

2800 S. River Road, Suite 170
Des Plaines, IL 60018-6092
Sales: 800-964-2793
Main web site: www.maxemail.com

MaxEmail allows you to receive faxes without owning a fax machine or dedicated fax line. You are assigned a unique fax number in one of 150 available area codes. Simply send faxes to your new fax number as usual. Faxes received are delivered to you via email as PDF attachments.

See advertisement on page 61 of this issue.

LC Technology International, Inc.

28100 US Highway 19
Suite 203
Clearwater, FL 33761
Sales: 866-603-2195 Toll Free or 727-449-0891 Local
Main web site: www.LC-Tech.com

LC Technology International, Inc. is a global leader in the data recovery market. With various software and services available, LC Technology offers advanced solutions to catastrophic data loss problems. Move your world forward today with products such as FILERECOVERY® Professional and PHOTORECOVERY® for Digital Media.

See advertisement on page 29 of this issue.

Lemke Software GmbH

Zum Rohkamp 5e
31228, Peine
Germany
Sales: 011 49 5171 72200
Main web site: www.lemkesoft.com

Open and save almost any picture file format. Edit and organize your pictures. Start a slide show. Automate your processing. And, and, and: GraphicConverter UB | X | Classic is your universal tool for all tasks related to digital photography.

See advertisement on page 24 of this issue.

Leo Impact

616 Corporate Way, Ste 2 #4000
Valley Cottage, NY 10989
Main web site: www.leoimpact.com

Leo Impact Security, Inc is a leading IT security research company providing specialized information extraction and IT security solutions.

See advertisement on page 9 of this issue.

MacTech Domains

PO Box 5200
Westlake Village, CA 91359
Sales: 805-494-9797
Main web site: www.mactech.com

Get your .COM or any other domain name here! Get a new domain name, transfer or renewal for as little as \$1.99* with each and every new, non-domain product you buy — no quantity limit! Every domain includes Complete Email (\$9.99/yr value!) and much more!

See advertisement on page 52 of this issue.

MacTech Magazine

PO Box 5200
Westlake Village, CA 91359
Sales: 877-622-8324
Main web site: www.mactech.com

The MacTech DVD - Volumes 1.01-27.03 is packed with more than ever before -- over 3300 articles from more than 300 issues (1984 - March 2011) written by almost 900 experts, all 29 issues of Apple's develop, 21 issues of FrameWorks magazine, all the source code, MacTech Viewer, working applications, full documentation, demos for techs, and more! The latest version includes all kinds of third-party applications, videos and demos.

See advertisement on page 49 of this issue.

Madcatz

7480 Mission Valley Rd. Ste 101
San Diego, CA 92108
Sales: (619) 683-9830
Main web site: www.madcatz.com

See advertisement on page 53 of this issue.

Matias Corporation

221 Narinia Crecent
New Market, ON L3X 2E1
Canada
Sales: 1-905-265-8844
Main web site: www.matais.ca

See advertisement on page 26 of this issue.

Media Sign Pro

PO BOX 50323
Irvine, CA 92619
Sales: 866.991.3713
Main web site: www.mdiasignpro.com

In our research of digital signage, we found that most applications were very expensive and very difficult to use. The other drawback we found was that most digital signage software required the use of two and sometimes three separate applications just to publish a project onto a display.

See advertisement on page 31 of this issue.

Microchip Technology Inc.

2355 W. Chandler Blvd
Chandler, AZ 85224
Main web site: www.microchip.com

All of your microchip needs

See advertisement on page 71 of this issue.

Micromat, Inc.

1007-B W. College Ave #333
Santa Rosa, CA 95401
Sales: 707-566-3831
Main web site: www.micromat.com

See advertisement on page 59 of this issue.

Microsoft

One Microsoft Way
Redmond, WA 98075
Sales: 800-MICROSOFT (642-7676)
Main web site: www.microsoft.com

Office 2008 for Mac: You'll build great looking documents in no time. Achieve more and simplify your workday. Enjoy productivity as you unleash your creative side with powerful, easy-to-use, intuitive tools.

See advertisement on page 23 of this issue.

Mindraven

318 E Marion Street
Aberdeen, WA 98520
Sales: 1.888.506.4582
Main web site: www.mindraven.com

Exceptional web hosting

See advertisement on page 60 of this issue.

MACTECH

Newsoft America

47102 Mission Falls Court
Fremont, CA 94539
Main web site: www.newsoftinc.com

Advanced scanning and copying software

See advertisement on page 20 of this issue.

OlympicControls Corp.

1250 Crispin Drive
Elgin, IL 60123
Sales: 847-742-3566
Main web site: www.occorp.com

BookEndz specializes in providing Apple customers with products that will enhance their experience with Apple Laptop Computers. Our goal is to provide quality Docking Stations and other accessory products as quickly as possible after Apple introduces new and updated laptop computers.

See advertisement on page 55 of this issue.

Parallels Inc.

500 SW 39th St., Ste 200
Renton, WA 98057
Sales: 425-282-6400
Main web site: www.parallels.com

Parallels Desktop for Mac is the award-winning desktop virtualization software that currently enables more than millions of Mac users to run Windows, Linux and other operating systems side-by-side with Mac OS X on any Intel-powered Mac — without rebooting!

See advertisement on page 70 of this issue.

Pioneer

1925 E. Dominguez St.
Long Beach, CA 90810
Main web site: www.pioneerelectronics.com

Pioneer is a leader in electronics products for car, home and business markets, respected for our role in introducing such innovations as consumer laser disc (1979), car CD player (1984), GPS car navigation (1990), DVD-Video player (1996), high definition plasma display (1997) and organic electroluminescent (OEL) display (1997).

See advertisement on inside front cover of this issue.

Prosoft Engineering, Inc.

303 Ray St.
Pleasanton, CA 94566
Sales: 877.477.6763
Main web site: www.prosofteng.com

Prosoft Engineering, Inc. is a software company focused on data recovery software and other utilities which help protect and manage your important data. Prosoft takes pride in its award-winning products, excellent customer service and ease of use. Prosoft has distribution partners throughout the Americas, Europe, Asia and Africa.

See advertisement on page 51 of this issue.

Quark Inc.

1800 Grant Street
Denver, CO 80203
Sales: 800.676.4575
Main web site: www.quark.com

Nearly 20 years ago, Quark led the revolution in desktop publishing. Today, Quark is revolutionizing publishing again by helping customers automate their publishing process to deliver accurate, relevant, and attractive communications anywhere.

See advertisement on page 15 of this issue.

REAL Software, Inc.

PO Box 162181
Austin, TX 78716
Sales: 512-328-7325
Main web site: www.realsoftware.com

REAL Studio is the easy-to-use integrated development environment that enables users at all levels to create powerful stand-alone, native applications quickly and easily, without having to learn a complex programming language.

See advertisement on page 37 of this issue.

Ruckus Wireless

880 West Maude Ave Suite 101
Sunnyvale, CA 94085
Sales: 650.265.4200
Main web site: www.ruckuswireless.com

Ruckus is seriously shaking up the Wi-Fi world with technology that lets enterprises and carriers take wireless where it's never gone before.

See advertisement on page 30 of this issue.

RunRev Limited

25a Thistle Street Lane South West
Edinburgh, Scotland EH2 1EW
United Kingdom
Sales: 44 (0) 845 219 8923
Main web site: www.runrev.com

The fastest way to develop iOS and Android Apps

See advertisement on page 1 of this issue.

SAINT

4720 Montgomery Lane
Ste 800
Bethesda, MD 20814
Sales: 800-596-2006
Main web site: www.saintcorporation.com

The SAINT team consists of information assurance experts with advanced degrees and technical certifications, including CISSP and PCI Qualified Security Assessors* (QSA). We are experts in safeguarding information systems and can help you effectively manage risk and protect your business-critical data.

See advertisement on page 25 of this issue.

Shure

32 Merchandise Mart Plaza, Ste 550
Chicago, IL 60654
Main web site: www.sparksmg.com

Legendary microphones. Cutting-edge wireless systems. Premium earphones and headphones. Welcome to Shure professional audio products.

See advertisement on page 2 of this issue.

Small Dog Electronics

1673 Main Street
Waitsfield, VT 05673
Sales: 800-511-MACS
Main web site: www.smalldog.com

Small Dog Electronics is an Apple Specialist, one of the larger Apple resellers in the US. We specialize in all things Macintosh, including Apple refurbished products.

See advertisement on inside back cover of this issue.

Sonnet Technologies, Inc.

8 Autry
Irvine, CA 92618-2708
Sales: 949.587.3500
Main web site: www.sonnettech.com

Fusion Storage Systems

See advertisement on page 8 of this issue.

Telestream

848 Gold Flat Road, Suite 1
Nevada City, CA 95959
Sales: 530-470-1300
Main web site: www.telestream.net

Capture the contents of your entire desktop at the same time as your video camera, microphone and computer's audio. Sophisticated editing tools allow you to create incredible screencasts in no time. The finished result is a QuickTime movie, ready for publishing to your website or blog.

See advertisement on page 46 of this issue.

Titan Radio

4840 S 35th Street
Phoenix, AZ 85040
Sales: 800-411-7080
Main web site: www.TitanRadio.com

See advertisement on page 11 of this issue.

TransTech Systems

12142 NE Sky Lane
Ste 130
Aurora, OR 97002
Sales: 888.843.3643
Main web site: www.ttsys.com

TransTech stocks a full line of ID badging and access control products.

See advertisement on page 27 of this issue.

Utilities4Less.com

1652 Cross Bridge Place
Thousand Oaks, CA 91362
Sales: (800) 906-8686
Main web site: www.utilities4less.com

Utilities4Less offers its clients a full range of communications services. Utilities4Less is committed to offering the best products available at the lowest possible prices.

See advertisement on page 34 of this issue.

VisiStat, Inc.

1875 S. Bascom Ave.
Ste 2445
Campbell, CA 95008
Sales: 310.237.5000
Main web site: www.visistat.com

VisiStat is an easy-to-use, real-time visitor tracking service — learn how people find your Website, where they come from, what they do, how long they stay and much more! Tracking visitors is traditionally technical and complex, and often raises more questions than it answers — not with VisiStat!

See advertisement on page 47 of this issue.

Watchman Monitoring

8568 Goodwood Blvd, Ste B
Baton Rouge, LA 70806
Main web site:
www.watchmanmonitoring.com

Watchman Monitoring enables you to provide proactive support, with a minimally intrusive solution. The Watchman Client software installs on your clients' computers and submits hourly status reports to your Watchman Server.

See advertisement on page 50 of this issue.

WheelsAway

20295 Cottagewood Rd.
Excelsior, MN 55331
Main web site: www.wheelsaway.com

Transport your iMac in style - safely and easily.

See advertisement on page 53 of this issue.

ZAGG Inc

3855 So. 500 W., Suite J
Salt Lake City, UT 84115
Sales: 888-940-ZONE (9663)
Main web site: www.shieldzone.com

ZAGG Inc is an industry leading manufacturer and provider of consumer products and services focusing on electronic handheld accessories. Our flagship product, the invisibleSHIELD, is a revolutionary film that has replaced the need for bulky cases and cheap screen protectors.

See advertisement on page 57 of this issue.

Product Guide

Listing by Category

Accessories

Black Magic Design: UltraStudio 3D, See page	17.
CableJive: cables/adaptors, See page	24.
CableJive: cables/adaptors, See page	41.
Gefen Inc.: DVI to MiniDisplayPort, See page	12.
LC Technology International, Inc.: PHOTORECOVERY®/FILERECOVERY®, See page	29.
Madcatz: Gaming Accessories, See page	53.
Matias Corporation: TactilePro, See page	26.
OlympicControls Corp.: BookEndz, See page	55.
Shure: Audio Products, See page	2.
Small Dog Electronics: SmallDog.com, See	IBC.
Sonnet Technologies, Inc.: RackMac Mini, See page	8.
Titan Radio: Dollar Radio, See page	11.
WheelsAway: WheelsAway, See page	53.
ZAGG Inc: Skins, See page	57.

Audio

Audioengine: Audio Speaker Systems, See page	28.
Shure: Audio Products, See page	2.

Backup Software

LC Technology International, Inc.: PHOTORECOVERY®/FILERECOVERY®, See page	29.
--	-----

Business Services

Background Backup, Inc.: Consulting, See page	18.
Brad Sniderman: Law Offices, See page	14.
Facet Corp.: IP PBX Phone System, See page	58.
IGC, Inc. / MaxEMail.com: maxemail.com, See page	61.
Media Sign Pro: Media Sign Pro, See page	31.
Ruckus Wireless: Ruckus Wireless, See page	30.
Utilities4Less.com: Long Distance Phone Service, See page	34.

Business Software

codefortytwo software: CrashPlan, See page	19.
Dartware LLC: InterMapper, See page	13.
FileWave (USA), Inc.: FileWave, See page	54.
Hansaworld: Business Management Software, See	BC.
Microsoft: Microsoft Office, See page	23.
Parallels Inc.: Parallels Desktop and Server, See page	70.

Communications, VoIP

Utilities4Less.com: Long Distance Phone Service, See page	34.
--	-----

Consumer Products

Audioengine: Audio Speaker Systems, See page	28.
Pioneer: Various Audio Products, See page	IFC.

CPUs and Upgrades

Microchip Technology Inc.: Microchips, See page	
Small Dog Electronics: SmallDog.com, See	IBC.

Data Recovery

DriveSavers Data Recovery: Data Recovery, See page	34.
Faronics: Deep Freeze, See page	33.
LC Technology International, Inc.: PHOTORECOVERY®/FILERECOVERY®, See page	29.

Databases

REAL Software, Inc.: REAL Studio Web Edition, See page	37.
VisiStat, Inc.: VisiStat, See page	47.

Developer Tools

RunRev Limited: LiveCode, See page	1.
--	----

Enterprise

Benchmark Email: Benchmark Email, See page	45.
codefortytwo software: CrashPlan, See page	19.
Dartware LLC: InterMapper, See page	13.
Faronics: Deep Freeze, See page	33.
IGC, Inc. / MaxEMail.com: maxemail.com, See page	61.
Intego, Inc.: Intego Virus Protection/Security, See page	
Microsoft: Microsoft Office, See page	23.
Parallels Inc.: Parallels Desktop and Server, See page	70.
REAL Software, Inc.: REAL Studio Web Edition, See page	37.
Ruckus Wireless: Ruckus Wireless, See page	30.
VisiStat, Inc.: VisiStat, See page	47.

Internet Services

Acquia, Inc.: drupal Hosting, See page	
HostGator: Hosting, See page	
IGC, Inc. / MaxEMail.com: maxemail.com, See page	61.
Ruckus Wireless: Ruckus Wireless, See page	30.
Utilities4Less.com: Long Distance Phone Service, See page	34.
VisiStat, Inc.: VisiStat, See page	47.
Watchman Monitoring: Watchman Monitoring, See page	50.

iPod

Small Dog Electronics: SmallDog.com, See	IBC.
ZAGG Inc: Skins, See page	57.

Mail Order/Retailer

Small Dog Electronics: SmallDog.com, See	IBC.
--	------

Miscellaneous

TransTech Systems: Identifying Solutions, See page	27.
--	-----

Mobile Device Mgmt

Fiberlink: MaaS360, See page	7.
------------------------------------	----

Multimedia, Graphics

Lemke Software GmbH: Graphic Converter, See page	24.
Quark Inc.: Quark, See page	15.
Telestream: Screenflow, See page	46.

Networking

Dartware LLC: InterMapper, See page	13.
CableJive: cables/adaptors, See page	41.
Faronics: Deep Freeze, See page	33.
Ruckus Wireless: Ruckus Wireless, See page	30.
Small Dog Electronics: SmallDog.com, See	IBC.
Watchman Monitoring: Watchman Monitoring, See page	50.

Peripherals

OlympicControls Corp.: BookEndz, See page	55.
---	-----

Printers and Output

Small Dog Electronics: SmallDog.com, See	IBC.
--	------

Productivity

Alice Dev Team: Utility Software, See page	14.
e3 Software: Direct mail, See page	21.
Faronics: Deep Freeze, See page	33.
Hansaworld: Business Management Software, See	BC.
IGC, Inc. / MaxEMail.com: maxemail.com, See page	61.
Lemke Software GmbH: Graphic Converter, See page	24.
Microsoft: Microsoft Office, See page	23.
Parallels Inc.: Parallels Desktop and Server, See page	70.
Quark Inc.: Quark, See page	15.

Repair Services

DriveSavers Data Recovery: Data Recovery, See page	34.
--	-----

Scripting Software

REAL Software, Inc.: REAL Studio Web Edition, See page	37.
--	-----

Security

LC Technology International, Inc.: PHOTORECOVERY®/FILERECOVERY®, See page	29.
Leo Impact: IT Security Research, See page	9.
OlympicControls Corp.: BookEndz, See page	55.
Prosoft Engineering, Inc.: Data Rescue Center, See page	51.
SAINT: Security Testing Tools, See page	25.

Server

Background Backup, Inc.: Consulting, See page	18.
codefortytwo software: CrashPlan, See page	19.
Mindraven: UBB.threads, See page	60.
VisiStat, Inc.: VisiStat, See page	47.

Server Software

codefortytwo software: CrashPlan, See page	19.
Mindraven: UBB.threads, See page	60.
VisiStat, Inc.: VisiStat, See page	47.

Storage

codefortytwo software: CrashPlan, See page	19.
--	-----

Training Related

ComputerTree: Training / Classes, See page	18.
Future Media Concepts: IT Training, See page	59.
iDeveloper TV: iDeveloper TV, See page	38.

Utilities

Alice Dev Team: Utility Software, See page	14.
Benchmark Email: Benchmark Email, See page	45.
Faronics: Deep Freeze, See page	33.
LC Technology International, Inc.: PHOTORECOVERY®/FILERECOVERY®, See page	29.
Lemke Software GmbH: Graphic Converter, See page	24.
Micromat, Inc.: TechToolPro, See page	59.
Newsoft America: PrestoBizCard, See page	30.
Parallels Inc.: Parallels Desktop and Server, See page	70.



PACKED!

The MacTech DVD - Volumes 1.01-27.03 is packed with more than ever before -- over 3300 articles from more than 300 issues (1984 - March 2011) written by almost 900 experts, all 29 issues of Apple's *develop*, 21 issues of *FrameWorks* magazine, all the source code, MacTech Viewer, working applications, full documentation, demos for techs, **and more!** The latest version includes all kinds of third-party applications, videos and demos.



See for yourself why *MacTech Magazine's* DVD is the best information source for Macintosh techs and developers. Search quickly through over 27 years of great information provided by *MacTech*. Information to save you time, and make your life easier.



**Call Toll Free 877-MACTECH,
Outside US/Canada: 805-494-9797
<http://www.mactech.com/dvd>**

Product Guide

Listing by Product

Audio Products	2
Audio Speaker Systems	28
Benchmark Email	45
BookEndz	55
Business Management Software	BC
cables/adaptors	24
Consulting	18
CrashPlan	19
Data Jack	41
Data Recovery	34
Data Rescue Center	51
Deep Freeze	33
Direct mail	21
Dollar Radio	11
Domain Registration	52
Drupal Hosting	35
DVI to MiniDisplayPort	12
FileWave	54
Gaming Accessories	53
Graphic Converter	24
Hosting	39
Identifying Solutions	27
iDeveloper TV	38
InterMapper	13
IP PBX Phone System	58
IT Security Research	9
IT Training	59
Law Offices	14
LiveCode	1
Long Distance Phone Service	34
MaaS360	7
maxemail.com	61
Media Sign Pro	31
Microchips	71
Microsoft Office	23
Parallels Desktop and Server	70
PHOTORECOVERY®/FILERECOVERY®	29
PrestoBizCard	20
Quark	15
RackMac Mini	8
REAL Studio Web Edition	37
Ruckus Wireless	30
Screenflow	46
Security Testing Tools	25
Skins	57
SmallDog.com	IBC
Smart Strips	10
TactilePro	26
TechToolPro	59
Training / Classes	18
UBB.threads	60
UltraStudio 3D	17
Utility Software	14
Various Audio Products	IFC
VisiStat	47
Watchman Monitoring	50
WheelsAway	53

Advertiser/Product Index

Ad Index by Company for: MacTech Mag 27.12 (December/2011)

Acquia, Inc.	35
Alice Dev Team	14
Audioengine	28
Background Backup, Inc.	18
Benchmark Email	45
BITS Limited	10
Black Magic Design	17
Brad Sniderman	14
CableJive	24
codefortytwo software	19
ComputerTree	18
Dartware LLC	13
Data Jack	41
DriveSavers Data Recovery	34
e3 Software	21
Facet Corp.	58
Faronics	33
Fiberlink	7
FileWave (USA), Inc.	54
Future Media Concepts	59
Gefen Inc.	12
Hansaworld	BC
HostGator	39
iDeveloper TV	38
IGC, Inc. / MaxEMail.com	61
LC Technology International, Inc.	29
Lemke Software GmbH	24
Leo Impact	9
MacTech Domains	52
MacTech DVD	68
Madcatz	53
Matias Corporation	26
Media Sign Pro	31
Microchip Technology Inc.	71
Micromat, Inc.	59
Microsoft	23
Mindraven	60
Newsoft America	20
OlympicControls Corp.	55
Parallels Inc.	70
Pioneer	IFC
Prosoft Engineering, Inc.	51
Quark Inc.	15
REAL Software, Inc.	37
Ruckus Wireless	30
RunRev Limited	1
SAINT	25
Shure	2
Small Dog Electronics	IBC
Sonnet Technologies, Inc.	8
Telestream	46
Titan Radio	11
TransTech Systems	27
Utilities4Less.com	34
VisiStat, Inc.	47
Watchman Monitoring	50
WheelsAway	53
ZAGG Inc.	57

Ad Index by Product for: MacTech Mag 27.12 (December/2011)

Audio Products • Shure	2
Audio Speaker Systems • Audioengine	28
Benchmark Email • Benchmark Email	45
BookEndz • OlympicControls Corp.	55
Business Management Software • Hansaworld	BC
cables/adaptors • CableJive	24
Consulting • Background Backup, Inc.	18
CrashPlan • codefortytwo software	19
Data Jack • Data Jack	41
Data Recovery • DriveSavers Data Recovery	34
Data Rescue Center • Prosoft Engineering, Inc.	51
Deep Freeze • Faronics	33
Direct mail • e3 Software	21
Dollar Radio • Titan Radio	11
Domain Registration • MacTech Domains	52
Drupal Hosting • Acquia, Inc.	35
DVI to MiniDisplayPort • Gefen Inc.	12
FileWave • FileWave (USA), Inc.	54
Gaming Accessories • Madcatz	53
Graphic Converter • Lemke Software GmbH	24
Hosting • HostGator	39
Identifying Solutions • TransTech Systems	27
iDeveloper TV • iDeveloper TV	38
InterMapper • Dartware LLC	13
IP PBX Phone System • Facet Corp.	58
IT Security Research • Leo Impact	9
IT Training • Future Media Concepts	59
Law Offices • Brad Sniderman	14
LiveCode • RunRev Limited	1
Long Distance Phone Service • Utilities4Less.com	34
MaaS360 • Fiberlink	7
MacTech DVD • MacTech Magazine	68
maxemail.com • IGC, Inc. / MaxEMail.com	61
Media Sign Pro • Media Sign Pro	31
Microchips • Microchip Technology Inc.	71
Microsoft Office • Microsoft	23
Parallels Desktop and Server • Parallels Inc.	70
PHOTORECOVERY®/FILERECOVERY® • LC Technology International, Inc.	29
PrestoBizCard • Newsoft America	20
Quark • Quark Inc.	15
RackMac Mini • Sonnet Technologies, Inc.	8
REAL Studio Web Edition • REAL Software, Inc.	37
Ruckus Wireless • Ruckus Wireless	30
Screenflow • Telestream	46
Security Testing Tools • SAINT	25
Skins • ZAGG Inc.	57
SmallDog.com • Small Dog Electronics	IBC
Smart Strips • BITS Limited	10
TactilePro • Matias Corporation	26
TechToolPro • Micromat, Inc.	59
Training / Classes • ComputerTree	18
UBB.threads • Mindraven	60
UltraStudio 3D • Black Magic Design	17
Utility Software • Alice Dev Team	14
Various Audio Products • Pioneer	IFC
VisiStat • VisiStat, Inc.	47
Watchman Monitoring • Watchman Monitoring	50
WheelsAway • WheelsAway	53

run windows
applications like
they were made
for your mac



Parallels Desktop® 7 for Mac
is the simplest and smartest
way to run Windows programs
on your Mac. **No rebooting
necessary.**

From business software to Windows Internet Explorer, PC games, and thousands more applications, Windows couldn't feel more natural on the computer you love.

Do more. Work in Mac OS X and Windows side-by-side, without rebooting. Use Mac OS X keyboard shortcuts and your iSight or FaceTime HD camera with both platforms.

Get more. Enjoy brilliant graphics, blazing performance, 7.1 Surround Sound and support for Mac OS X Lion's amazing features—like Launchpad, Mission Control and gestures—even in Windows.



Connect. Access your computer's Windows and Mac applications with the Parallels Mobile App for iPhone and iPad.

find out more!



Go ahead. Bring Windows to your Mac. Get your hands on the #1 choice¹ of Mac users worldwide. Learn more at www.parallels.com/desktop.

MOST TESTED, TRUSTED
AND TALKED ABOUT²



In our line of work things like github, StackOverflow, Twitter and blogs make this easier than when I was coming up. You break into things by being noticed. You get noticed by being interesting. You're interesting when you've done something remarkable. Concentrate on making something remarkable, then share it with people. It doesn't even matter if the idea itself is good, the execution can get you noticed. Think of it as being an apprentice and you're studying under a stable of masters who's work you admire and, for one of the first times in history, it's incredibly easy to reach out to them for comment. Work to impress those whose work impresses you. That's been my approach coming up and I still operate that way. I think once you stop doing that you've stopped being personally invested and that's when you stop improving.

That's not to say that simply working hard gets you anywhere, you need a degree of good fortune and support. With the community as large and as strong as it is today, and with communication being as simple as it is, I think there's a better chance now of benefiting from fortunate circumstances than we had ten or twenty years ago.

What's the coolest tech thing you've done using OS X?

I can't help but read this question as a matter of perspective. One of the very first applications I wrote for Rhapsody (early OS X) was a text editor that supported rich text editing with embedded images. It was only a handful of lines of code. It blew me away; the frameworks were decades ahead of anything else I'd seen. But that's not impressive now, and it's not something I did from scratch. To me it was one of the coolest tech things because it was one of those moments where you realize you're standing on the shoulder of giants and you're standing on the shoulder of a giant that all the other giants make fun of because he's just far, far taller than the rest of them.

On iOS I've done a couple of projects I really enjoyed. I worked on a paint bucket algorithm (and did some exploratory background removal work) that was based on OpenCL to do the pixel processing. I had a really good time doing that, and I'm grateful for that opportunity. I did another fun little project when I wrote a signature view that had the ink and stroke width vary according to how fast the finger was moving. I'm not sure either of these are really something anyone would say, "wow, that's cool!" but they were a blast to work on and interesting problems.

Ever?

Ever? I wrote the audio system for the engine that powered a bunch of games on the PSP back when I was in the video games industry. The coolest tech thing I did you'd never notice — I wrote the music streaming code that would pull audio from the UMD disk and made sure it wouldn't skip. We also had to support jumping to different kinds of music as the gameplay dictated. The reason this was "cool" was because there's an up to two second seek time for the UMD in the original PSP which meant juggling all sorts of limits, from memory buffer sizes, to predictive pre-reads to level audio design. The good news is that after all that work the player never noticed that it was fancy, it was just right.

MACTECH

Anything that you consider indispensable for your work?

Twitter, IRC and iChat. Being able to talk to other people is invaluable. Both when you've got a question and when you just want to complain about being stuck. People are great.

Where can we see a sample of your work?

Well, I mostly work on a consultant basis but I wrote the engine that is the basis for the Tap Tap Revenge series of games and related spin offs. They've become quite popular. On the Mac I wrote the first version of RadioShift for Rogue Amoeba. My code is also in a lot of other apps, mostly on iOS now but also some popular Mac apps.

The next way I'm going to impact the Mac/iOS/Apple universe is:

My friend Chris Parrish and will soon be releasing a new Mac app that aims to enable concise visual communication. Also, I just helped organize the Cingleton Symposium that gathered a group of great speakers to provide a community forum to discuss the massive technological and market changes that have come about over this past year. It was well received and we're expecting to do another next year.

MI

If you or someone you know belongs in the MacTech Spotlight, let us know! Send details to editorial@mactech.com

Microchip's development platforms let you easily design & develop an application or accessory for iPad, iPhone and iPod today!



The mobile device marketplace offers the promise of entirely new uses for ever smarter devices. Let Microchip's development platforms help jump start your idea for the next great accessory.

Our kits enable fast and easy development of accessories based on Microchip's large portfolio of 16-bit and 32-bit PIC® microcontrollers (MCUs) and dsPIC® Digital Signal Controllers.

Available Development Kits:

- Digital Audio Development Kit for iPod and iPhone
- 8-bit PIC MCU Accessory Development Kit for iPod and iPhone
- 16/32-bit PIC MCU Accessory Development Kit for iPod and iPhone
- Microchip's MFi Library for iPod and iPhone



Development resources are also available for Android. For more information, visit: www.microchip.com/smartphone

Get Started:

- Ask your Microchip Sales representative to schedule a demonstration
- Enroll in Apple's "Made for iPod" licensing program
- Purchase a development kit through Apple's Made-for-iPod program
- Get Started at: <http://www.microchip.com/MFi>

www.microchip.com/MFi

MICROCHIP

Guy English

<http://kickingbear.com>

What's your company, and what do you do?

Mostly consultant to the stars but I've just recently founded Aged & Distilled with my friend Chris Parrish.

Specifically, I write software. As for which aspect, for the most part, I do whatever I'm hired to do. I've been lucky to have had the opportunity to work in a lot of different fields, from UI code to scripting bridges to audio processing and graphics code. I'm mostly a graphics guy though, I suppose, having done a lot of work with OpenGL while I was in the games industry.

How long have you been doing what you do?

That's a tough question. I started programming when I was seven and I sort of see where I'm at now as the natural progression from that. Along the way my tastes have changed and I've been fortunate enough to be able to make a living out of it. I started getting into writing software for the Mac when I bought my beige G3 back in '97 right after the NeXT acquisition. I'd been a huge fan of NeXTStep and was really excited to get my hands on what NeXT and Apple would come up with. I kept writing software for myself on the Mac until I finally gave up my day job in the games business and joined Rogue Amoeba in about 2005 and had the opportunity to make a living out of it. Once the iPhone came out I loved working with it and eventually left Rogue Amoeba and headed out on my own doing coding for hire work. So, 2008-ish.

What was your first computer?

A beautiful Apple II+ with the awesome 80 column card and a push switch on a wire that would kick that sucker into turbo. Which would be a bad thing to do on a hot summer day because it'd overheat in minutes. Loved that computer.

Are you Mac-only, or a multi-platform person?

Actually, funny story—I never liked the Mac that much. After my Apple II+ (loved that computer) I moved over to PCs and started to get really into the intricacies of programming the graphics hardware. Not the audio hardware because that was a beeper that made that classic annoying PC beep when they'd start up. But the graphic cards were pretty cool and it was fun to play with the registers and mess around at that sort of low level. That was during the DOS and early Mac days. The next OS I went for was OS/2 which, I'll contend to this day, was a great OS with a lot of concepts that were really remarkable and ahead of their time. That didn't end so well though.

So I found myself on Windows NT 4.0 in the late nineties when Apple bought NeXT. By that point I had various FreeBSD machines doing my internet routing and such and I was a huge fan of NeXTStep, it was just always so expensive! So I bought a Mac.



It's cost me about \$700 to buy the OS/2 development tools so I sort of saw buying this beige G3 (AV w/ 266Mhz!) as an investment in a tool. It shipped with OS8 but Apple promised that Rhapsody would run on it once released. Good enough for me.

I could not believe how primitive and advanced the Mac was all at the same time. Holding down scrollbars in IE would stall downloads. Coming from OS/2 and NT that's just totally insane. On the other hand, there was something about it that was just right. Siracusa spent a lot of words around that time trying to define it in terms of the spacial Finder and the way everything fit together but, really, I think it just felt right.

I've been a Mac only guy since then, especially once OS X came out and it had it all for me — great UI, UNIX, amazing development environment and tools, and really remarkable apps from smaller developers which I found really inspirational.

What is the advice you'd give to someone trying to get into this line of work today?

I'm often asked to recommend books for someone looking to break into this racket. The go to answer is the Aaron Hillegass' *Cocoa Programming for Mac OS X*. It's been a while since I've looked at a revision but Aaron is a great teacher. If you want to get a racing head start, consider attending Aaron's Big Nerd Ranch. I've heard nothing but great things about it and some brainy friends of mine teach there so it's bound to be good.

At a less specific level though my advice, to anyone who wants to break into something, is simply to do it. Want to make something, then make it. Then do it again. And again. Share what you've made with people who make things you admire and then go back and make something more.

Continues on page 71

Every business needs a Small Dog to help it grow.

Small Dog Electronics, New England's largest Apple Specialist, has the expertise, products and services to support your technology needs.

- ☒ Everything Apple®, peripherals and accessories
- ☒ Remote training and technical support
- ☒ iOS integration and deployment
- ☒ Cross-platform solutions
- ☒ Mobile device management solutions



**Small Dog
Electronics**

Always By Your Side

800-511-MACS



Run ~~Find~~ a Restaurant

Ordering | Kitchen Printing | Reporting

Tabs opened from graphical table view. Fast, accurate orders, including special requirements. Wireless printing to kitchen and for final bill. Improve your customer service with Restaurant by HansaWorld, for iPad.



Books
by HansaWorld

Enterprise
by HansaWorld